

(19) 日本国特許庁 (J P)

## (12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平 10-191392

(43) 公開日 平成 10 年 (1998) 7 月 21 日

(51) Int. Cl.	識別記号	庁内整理番号	F I	技術表示箇所
H04N 11/20			H04N 11/20	
G06T 3/40			1/387	101
H04N 1/387	101		3/223	
3/223			5/66	B
5/66			9/64	Z

審査請求 未請求 請求項の数 26 O L (全 43 頁) 最終頁に続く

(21) 出願番号 特願平 9-44471  
 (22) 出願日 平成 9 年 (1997) 2 月 27 日  
 (31) 優先権主張番号 特願平 8-287173  
 (32) 優先日 平 8 (1996) 10 月 29 日  
 (33) 優先権主張国 日本 (J P)

(71) 出願人 000002185  
 ソニー株式会社  
 東京都品川区北品川 6 丁目 7 番 35 号  
 (72) 発明者 加納 護  
 東京都品川区北品川 6 丁目 7 番 35 号 ソ  
 ニー株式会社内  
 (72) 発明者 黒川 益義  
 東京都品川区北品川 6 丁目 7 番 35 号 ソ  
 ニー株式会社内  
 (72) 発明者 岩瀬 清一郎  
 東京都品川区北品川 6 丁目 7 番 35 号 ソ  
 ニー株式会社内  
 (74) 代理人 弁理士 小池 晃 (外 2 名)

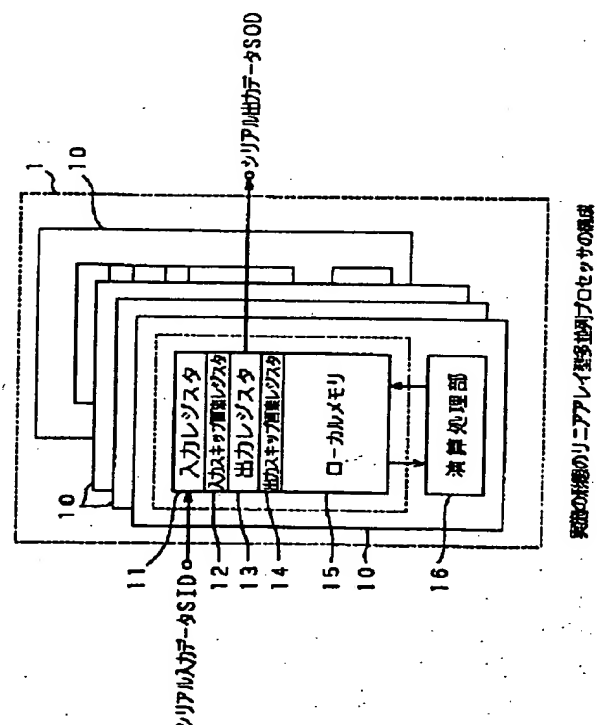
最終頁に続く

(54) 【発明の名称】 画像信号処理装置

## (57) 【要約】

【課題】 任意のクロマフォーマットに対応した任意比率の画素数変換及び走査線数を可能にする。

【解決手段】 SIMD 制御のリニアアレイ型多並列プロセッサ 1 を使い、ソフトウェア処理だけで任意比率の画素数変換処理を行う。このとき、4:4:4 フォーマットの場合は、輝度及びクロマに対してそれぞれ同一の画素数変換を行う。4:2:2 フォーマットの場合は、輝度信号に対してキュービック補間を行い、クロマ信号に対して直線補間を行って画素数変換を行うか、または、4:4:4 フォーマットに変換して上記同様の画素数変換を行う。4:1:1 フォーマットの場合は、4:2:2 フォーマットに変換して上記同様の画素数変換処理を行うか、または 4:2:2 フォーマットに変換したものを更に 4:4:4 に変換して上記同様の画素数変換処理を行う。



## 【特許請求の範囲】

【請求項 1】 デジタル化された 2 次元画像の 1 次元方向の各画素に対応して配置すると共に上記 1 次元方向の各画素データが時系列に順次入力する複数の要素プロセッサと、各要素プロセッサを共通に制御するための制御手段とを備える画像信号処理装置であって、

上記各要素プロセッサは、輝度及び色差の画素データを一時的に保存する一時保存手段と、輝度及び色差の入力画素データを格納して上記一時保存手段に転送する入力画素データ格納手段と、少なくとも輝度の画素の属性を表す画素属性情報を格納する画素属性情報格納手段と、輝度及び色差の画素データをスキップさせる画素スキップ情報を格納する画素スキップ情報格納手段と、上記画素属性情報に基づいて上記輝度及び色差の入力画素データ又は近傍の要素プロセッサの輝度及び色差の画素データを用いた所定の演算を行い得られた画素データを上記一時保存手段に保存する算術演算手段と、上記一時保存手段から取り出された輝度及び色差の画素データを格納して出力する出力画素データ格納手段と、を有してなることを特徴とする画像信号処理装置。

【請求項 2】 上記各要素プロセッサでは、4 : 4 : 4 フォーマットの輝度及び色差の画素データに対してそれぞれ同一の処理を行うことを特徴とする請求項 1 記載の画像信号処理装置。

【請求項 3】 上記各要素プロセッサの上記算術演算手段では、4 : 2 : 2 フォーマットの上記輝度の画素データに対して近傍 4 画素の値を用いた補間演算を行い、上記色差の画素データに対して近傍画素の値を用いた直線補間演算を行うことを特徴とする請求項 1 記載の画像信号処理装置。

【請求項 4】 上記各要素プロセッサは、4 : 2 : 2 フォーマットの色差の画素データを 4 : 4 : 4 フォーマットに変換するためのフォーマット変換手段を備え、各要素プロセッサは、上記 4 : 4 : 4 フォーマットの輝度及び色差の画素データに対して同一の処理を行うことを特徴とする請求項 1 記載の画像信号処理装置。

【請求項 5】 上記各要素プロセッサは 4 : 1 : 1 フォーマットの色差の画素データを 4 : 2 : 2 フォーマットに変換するためのフォーマット変換手段を備え、上記各要素プロセッサの上記算術演算手段では、上記 4 : 2 : 2 フォーマットの輝度の画素データに対して近傍 4 画素の値を用いた補間演算を行い、上記色差の画素データに対して近傍画素の値を用いた直線補間演算を行うことを特徴とする請求項 1 記載の画像信号処理装置。

【請求項 6】 上記各要素プロセッサは、4 : 1 : 1 フォーマットの色差の画素データを 4 : 4 : 4 フォーマットに変換するためのフォーマット変換手段を備え、各要素プロセッサでは、上記 4 : 4 : 4 フォーマットの輝度及び色差の画素データに対して同一の処理を行うことを特徴とする請求項 1 記載の画像信号処理装置。

【請求項 7】 上記各要素プロセッサの入力画素データ格納手段は、上記画素スキップ情報格納手段が格納する上記画素スキップ情報に基づいて、上記入力画素データを離散的或いは連続的に格納することを特徴とする請求項 1 記載の画像信号処理装置。

【請求項 8】 上記各要素プロセッサの出力画素データ格納手段は、上記画素スキップ情報格納手段が格納する画素スキップ情報に基づいて、上記一時記憶手段からの画素データを離散的或いは連続的に格納することを特徴とする請求項 1 記載の画像信号処理装置。

【請求項 9】 上記制御手段は、上記出力画素データ格納手段から出力する画素データのレートを、上記入力画素データ格納手段に入力する画素データのレートとは独立に制御することを特徴とする請求項 1 記載の画像信号処理装置。

【請求項 10】 上記各要素プロセッサは、上記画素属性情報を生成する画素属性情報生成手段を備えることを特徴とする請求項 1 記載の画像信号処理装置。

【請求項 11】 デジタル化された 2 次元画像の 1 次元方向の各画素に対応して配置すると共に上記 1 次元方向の各画素データが時系列に順次入力する複数の要素プロセッサと、各要素プロセッサを共通に制御するための制御手段とを備える画像信号処理装置であって、1 次元方向に配置された要素プロセッサへの入力部及び／又は出力部にて上記 2 次元画像データを格納するための 2 次元画像データ格納手段を備え、

上記要素プロセッサは、輝度及び色差の画素データを走査線毎に一時的に保存する走査線データ一時保存手段と、輝度及び色差の入力走査線データを格納して上記走査線データ一時保存手段に転送する入力走査線データ格納手段と、少なくとも輝度の画素からなる走査線の属性を表す走査線属性情報を格納する走査線属性情報格納手段と、上記走査線データをスキップさせる走査線スキップ情報を格納する走査線スキップ情報格納手段と、上記走査線属性情報に基づいて走査線データ又は近傍の要素プロセッサの走査線データを用いた所定の演算を行い得られた走査線データを上記走査線データ一時保存手段に保存する垂直方向算術演算手段と、上記走査線データ一時保存手段から取り出された走査線データを格納して出力する出力走査線データ格納手段とを有し、

上記走査線スキップ情報に基づいて、上記 2 次元画像データ格納手段へのデータ格納或いは上記 2 次元画像データ格納手段からのデータ取り出しを制御することにより、走査線数の拡大或いは縮小を行うことを特徴とする画像信号処理装置。

【請求項 12】 上記要素プロセッサは、上記走査線属性情報に基づいて、上記 2 次元画像データ格納手段へのデータ格納と上記 2 次元画像データ格納手段からのデータ取り出しとを制御する上記スキップ走査線情報を決定するスキップ走査線算術演算手段を備えることを特徴と

する請求項 1 1 記載の画像信号処理装置。

【請求項 1 3】 上記走査線スキップ情報格納手段は上記走査線スキップ情報を上記走査線数の拡大用と縮小用とで独立に格納し、上記走査線属性情報格納手段は上記走査線属性情報を上記走査線数の拡大用と縮小用とで独立に格納することで、縮小から拡大までの走査線数変換をリアルタイムに行うことを特徴とする請求項 1 1 記載の画像信号処理装置。

【請求項 1 4】 上記要素プロセッサは、輝度及び色差の画素データを格納して一時的に保存する画素データ一時保存手段と、輝度及び色差の入力画素データを格納して上記画素データ一時保存手段に転送する入力画素データ格納手段と、少なくとも輝度の画素の属性を表す画素属性情報を格納する画素属性情報格納手段と、輝度及び色差の入力画素データをスキップさせる入力画像スキップ情報と、輝度及び色差の出力画素データをスキップさせる出力画素スキップ情報とをそれぞれ格納する画素スキップ情報格納手段と、上記画素属性情報に基づいて上記輝度及び色差の入力画素データ又は近傍の要素プロセッサの輝度及び色差の画素データを用いた所定の演算を行い得られた画素データを上記画素データ一時保存手段に保存する水平方向算術演算手段と、上記画素データ一時保存手段から取り出された輝度及び色差の画素データを格納して出力する出力画素データ格納手段とを有し、画素数の拡大或いは縮小をも行うことを特徴とする請求項 1 1 記載の画像信号処理装置。

【請求項 1 5】 上記画素属性情報と上記入力画素スキップ情報と上記出力画素スキップ情報とを 1 走査線時間毎に設定して、リアルタイムに拡大又は縮小の画素数変換を行うことを特徴とする請求項 1 4 記載の画像信号処理装置。

【請求項 1 6】 上記制御手段は、上記要素プロセッサを入力画像の一走査線毎に走査線数変換用と画素数変換用に切り替え制御することを特徴とする請求項 1 4 記載の画像信号処理装置。

【請求項 1 7】 上記画素属性情報と上記画素スキップ情報と上記走査線スキップ情報とをそれぞれ独立に設定することにより、水平方向と垂直方向とでそれぞれ独立に拡大或いは縮小を行うことを特徴とする請求項 1 4 記載の画像信号処理装置。

【請求項 1 8】 上記要素プロセッサの入力部に配置された 2 次元画像データ格納手段の入力画像データレートを、上記要素プロセッサの出力部に配置された 2 次元画像データ格納手段の出力画像データレートとは独立に制御することを特徴とする請求項 1 1 記載の画像信号処理装置。

【請求項 1 9】 上記各要素プロセッサでは、4 : 4 : 4 フォーマットの輝度及び色差の画素データに対してそれぞれ同一の処理を行うことを特徴とする請求項 1 4 記載の画像信号処理装置。

【請求項 2 0】 上記各要素プロセッサの上記水平方向算術演算手段では、4 : 2 : 2 フォーマットの上記輝度の画素データに対して近傍 4 画素の値を用いた補間演算を行い、上記色差の画素データに対して近傍画素の値を用いた直線補間演算或いは最近傍画素の値をそのまま補間する処理を行うことを特徴とする請求項 1 4 記載の画像信号処理装置。

【請求項 2 1】 上記各要素プロセッサの上記垂直方向算術演算手段では、4 : 2 : 2 フォーマットの上記輝度の画素データに対して近傍 4 ラインの値を用いた補間演算を行い、上記色差の画素データに対して近傍ラインの値を用いた直線補間演算或いは最近傍画素の値をそのまま補間する処理を行うことを特徴とする請求項 1 4 記載の画像信号処理装置。

【請求項 2 2】 上記各要素プロセッサは、4 : 2 : 2 フォーマットの色差の画素データを 4 : 4 : 4 フォーマットに変換するためのフォーマット変換手段を備え、各要素プロセッサは、上記 4 : 4 : 4 フォーマットの輝度及び色差の画素データに対して同一の処理を行うことを特徴とする請求項 1 4 記載の画像信号処理装置。

【請求項 2 3】 上記各要素プロセッサは、4 : 1 : 1 フォーマットの色差の画素データを 4 : 2 : 2 フォーマットに変換するためのフォーマット変換手段を備え、上記各要素プロセッサの水平方向算術演算手段では、上記 4 : 2 : 2 フォーマットの輝度の画素データに対して近傍 4 画素の値を用いた補間演算を行い、上記色差の画素データに対して近傍画素の直線補間或いは最近傍画素の値をそのまま補間する処理を行うことを特徴とする請求項 1 4 記載の画像信号処理装置。

【請求項 2 4】 上記各要素プロセッサの入力画素データ格納手段は、上記画素スキップ情報格納手段が格納する上記画素スキップ情報に基づいて、上記入力画素データを離散的或いは連続的に格納することを特徴とする請求項 1 4 記載の画像信号処理装置。

【請求項 2 5】 上記出力画素データ格納手段から出力する画像のレートを、上記入力画素データ格納手段に入力する画素データのレートとは独立に制御することを特徴とする請求項 1 4 記載の画像信号処理装置。

【請求項 2 6】 上記各要素プロセッサは、上記画素属性情報を生成する画素属性情報生成手段と、上記走査線属性情報を生成する走査線属性情報生成手段とを備えることを特徴とする請求項 1 4 記載の画像信号処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、画素数変換処理や走査線変換等の画像信号処理を行う画像信号処理装置に関する。

【0002】

【従来の技術】 近年になって、半導体技術、半導体の処理スピード性能の向上により、映像信号のデジタル信

号処理が行われるようになってきた。また、最近では画像表示装置も従来のブラウン管に代わり、LCD (Liquid Crystal Display: 液晶ディスプレイ) 表示装置やプラズマディスプレイ装置等に代表される固定画素表示装置が広く普及しつつある。

【0003】また、最近では、いわゆるNTSC (National Television System Committee) 信号、PAL (Phase Alternation by Line) 信号などの標準テレビジョン放送方式のみならず、HDTV (High Definition Television) 信号や、VGA (Video Graphics Array) 信号、SVGA (Super VGA) 信号、XVGA (extended VGA) 信号など様々なフォーマットの信号を表示できることが求められている。

【0004】これら様々なフォーマットでは、それぞれ扱う画素数がまちまちである。このようなそれぞれ画素数が異なる各種のフォーマットの映像信号を表示する場合、上記ブラウン管等のアナログ表示デバイスであれば、1走査線時間当たりの画素数に応じて電子ビームの偏向速度を変えてやれば済む。

【0005】しかし、上記固定画素表示装置においては、扱える画素数が固定しているため、上述のブラウン管の場合のような従来のアナログ技術は使えない。そのため、これら様々なフォーマットの信号を上述のような固定画素表示装置に対して表示させるためには、デジタル信号処理による任意の画素数変換、或いは走査線数変換が不可欠である。

【0006】上述の画素数変換処理について、以下にその概要を説明する。

【0007】画素数変換処理とは、1走査線期間において入力画素数に対して出力画素数を所望の画素数に増減する処理であり、例えば入出力のサンプリング周波数が同じであるとした場合に、画素数を増加させたならば入力画像の拡大処理 (拡大画素数変換処理) となり、逆に画素数を減少させたならば入力画像の縮小処理 (縮小画素数変換処理) となる。別の言い方として、画素数ではなく、入出力の画素と画素のサンプリングという点でとらえれば、元々あるサンプリング点のデータから、元々のサンプリング位置とは異なる点のデータを作り出すことになり、この異なる点のデータを、入力された画素データから補間により補間画素を生成することに相当する。

【0008】この補間方法には様々な方法があり、大きく分けて以下の3つの方法が知られている。

【0009】1. ニアリストネイバー補間法

この方法は入力画素の画素データから画素数変換後の画素の位置に最も近い位置にあるデータを拾い出すやり方であり、ハードウェア構成は極めて簡単なロジック演算で実現できる。しかし、変換後の画質はかなり悪化する。縮小時は細い線が消えてしまったり、小さい図形がゆがみ、拡大時には周辺部にギザギザが発生したりす

る。

【0010】2. バイリニア補間法

この方法は、入力画素の画素データから画素数変換後の画素の位置に最も近い位置にある2点のデータを拾い出し、その2点のデータから線形補間するというもので、上記ニアリストネイバー補間法よりは画質の劣化が少ない。しかし、2:1以下に縮小するといわれるピクセルドロップアウトという現象が発生し、画質は一気に悪化する。また、この手法は、緩やかなローパスフィルタを施していることになるため、エッジ部分は特にそうであるが、全体的にぼけた画質になる。また、ハードウェア的にはニアレストネイバー補間法に比較すれば一気に複雑になる。

【0011】3. フィルタスイッチング補間法

この方法は、高画質の画像信号処理に用いられ、サイズの変換比に合わせたFIRフィルタ (例えばフィニットレスポンスフィルタ等) のデジタルフィルタを使って変換する方法である。しかし、この方法をハードウェアで構成しようとする、飛躍的に複雑で、大規模なものになるため、バイリニア補間法で行うのがほとんどである。

【0012】以下では、フィルタスイッチング補間法の一例として、後述するキュービック補間関数を用いた補間法について説明する。

【0013】先ず、例えば入力画素2個に対して出力画素3個を作り出すような2:3拡大画素数変換の原理について説明する。

【0014】図36には上記2:3拡大画素数変換の原理を説明するための図を示す。なお、この図36では、各入力画素の値をそれぞれ $R_{1,1}$ ,  $R_{1,2}$ ,  $R_{1,3}$ ,  $R_{1,4}$ ,  $R_{1,5}$ ,  $R_{1,6}$ , ... とし、各出力画素の値をそれぞれ $Q_{1,1}$ ,  $Q_{1,2}$ ,  $Q_{1,3}$ ,  $Q_{1,4}$ ,  $Q_{1,5}$ ,  $Q_{1,6}$ , ... として表している。また、図36の中の $P_{1,1}$ ,  $P_{1,2}$ ,  $P_{1,3}$ ,  $P_{1,4}$ , ... は、入力画素と出力画素の位相のずれ (位相情報) を表している。

【0015】ここで、上記2:3拡大画素数変換においては、この図36のように入力画素2個に対して出力画素3個を作り出すようにしており、入力画素と出力画素の関係は、出力画素の値がその近傍の入力画素から計算されるという関係になっている。上記出力画素を生成するための上記近傍範囲としてどのくらいまでの範囲を使用するか、或いは入力画素から補間により出力画素を計算する際の各係数の値としてどのような係数値を使用するかなどにより、様々な補間法が存在するが、以下の説明では、上記近傍範囲として4点 (4画素) 分の範囲から補間するキュービック補間を例に挙げている。

【0016】上記キュービック補間にて使用されるキュービック補間関数 $Cub(x)$ を図37に示し、その関数式を式(1)に示す。ただし、式(1)に示されるキュービック補間関数の横軸は原画像をデジタル信号にサンプリングする際のサンプリング間隔で正規化されてい

るものとする。

[0017]

$$\begin{aligned} \text{Cub}(x) &= |x|^3 - 2|x|^2 + 1 & (|x| \leq 1 \text{ の時}) \\ \text{Cub}(x) &= -|x|^3 + 5|x|^2 - 8|x| + 4 & (1 < |x| \leq 2 \text{ の時}) \\ \text{Cub}(x) &= 0 & (2 < |x| \text{ の時}) \end{aligned}$$

... (1)

拡大画素数変換の場合、各出力画素の補間値は、入力画素のサンプリング値とキュービック関数との畳み込み演

算で表され、出力画素の補間値は次式(2)のように表すことができる。

$$\begin{aligned} Q_{1,1} &= \text{Cub}(x_{1,1}) * R_{1,1} + \text{Cub}(x_{1,2}) * R_{1,2} + \text{Cub}(x_{1,3}) * R_{1,3} + \text{Cub}(x_{1,4}) * R_{1,4} \\ Q_{1,2} &= \text{Cub}(x_{2,1}) * R_{1,1} + \text{Cub}(x_{2,2}) * R_{1,2} + \text{Cub}(x_{2,3}) * R_{1,3} + \text{Cub}(x_{2,4}) * R_{1,4} \\ Q_{1,3} &= \text{Cub}(x_{3,1}) * R_{1,1} + \text{Cub}(x_{3,2}) * R_{1,2} + \text{Cub}(x_{3,3}) * R_{1,3} + \text{Cub}(x_{3,4}) * R_{1,4} \\ &\dots (2) \end{aligned}$$

この式(2)の各係数Cub(x)は前記キュービック補間関数から計算される値であり、これは、求めるべき出力画素が入力画素に対して、どれだけずれているかを示す位相から計算される。例えば、図36に示す2:3の拡大画素数変換の場合、上記Q<sub>1</sub>の出力画素の位相はその近傍の入力画素(例えばR<sub>1</sub>の入力画素)の位相と一致しているのでその位相情報P<sub>1</sub>はゼロとなり、同様に上記Q<sub>1,1</sub>の出力画素の位相はその近傍の入力画素

(例えばR<sub>1</sub>の入力画素)の位相から2/3ずれているのでその位相情報P<sub>1</sub>は2/3となり、上記Q<sub>1,2</sub>の出力画素の位相はその近傍の入力画素(例えばR<sub>1,1</sub>の入力画素)の位相から1/3ずれているのでその位相情報P<sub>1</sub>は1/3となるので、上記式(2)は式(3)のように書き換えることができる。

[0018]

$$\begin{aligned} Q_1 &= \text{Cub}(-1) * R_{1,1} + \text{Cub}(0) * R_1 + \text{Cub}(1) * R_{1,1} + \text{Cub}(2) * R_{1,2} \\ Q_{1,1} &= \text{Cub}(-5/3) * R_{1,1} + \text{Cub}(-2/3) * R_1 + \text{Cub}(1/3) * R_{1,1} + \text{Cub}(4/3) * R_{1,2} \\ Q_{1,2} &= \text{Cub}(-4/3) * R_1 + \text{Cub}(-1/3) * R_{1,1} + \text{Cub}(2/3) * R_{1,2} + \text{Cub}(5/3) * R_{1,3} \\ &\dots (3) \end{aligned}$$

上記Cub(x)及び入力画素の各値R<sub>1,1</sub>、R<sub>1</sub>、R<sub>1,1</sub>、R<sub>1,2</sub>はそれぞれ既知の値であるので、この式(3)から各出力画素の補間データが計算できる。例えば、上記

Q<sub>1</sub>の出力画素に限って言えば、前記式(1)より、Cub(-1)=0、Cub(0)=1、Cub(1)=0、Cub(2)=0なので、

$$Q_1 = 0 * R_{1,1} + 1 * R_1 + 0 * R_{1,1} + 0 * R_{1,2} = R_1 \quad \dots (4)$$

となり、入力画素の値そのものとなる。

表している。また、図38の中のP<sub>1</sub>、P<sub>2</sub>、P<sub>1</sub>、...も、入力画素と出力画素の位相のずれ(画素位相情報)を表している。

[0019]以上、2:3拡大画素数変換の場合を例にとって説明したが、任意の拡大比率でも同様であり、出力画素の位相さえわかれば、その位相によって式(1)からキュービック関数の各係数を求め、補間画素近傍の入力画素4点と畳み込み演算を行えばよい。

[0020]次に、例えば入力画素3個に対して出力画素2個を作り出すような3:2縮小画素数変換の原理について説明する。

[0021]図38には上記3:2縮小画素数変換の原理を説明するための図を示す。なお、この図38においても前記図36と同様に、各入力画素の値をそれぞれR<sub>1,1</sub>、R<sub>1</sub>、R<sub>1,1</sub>、R<sub>1,2</sub>、R<sub>1,3</sub>、...とし、各出力画素の値をそれぞれQ<sub>1</sub>、Q<sub>1,1</sub>、Q<sub>1,2</sub>、...として

$$\begin{aligned} Q_1 &= \text{Cub}(x_{1,1}) * R_{1,1} + \text{Cub}(x_{1,2}) * R_1 + \text{Cub}(x_{1,3}) * R_{1,1} + \text{Cub}(x_{1,4}) * R_{1,2} \\ Q_{1,1} &= \text{Cub}(x_{2,1}) * R_1 + \text{Cub}(x_{2,2}) * R_{1,1} + \text{Cub}(x_{2,3}) * R_{1,2} + \text{Cub}(x_{2,4}) * R_{1,3} \\ &\dots (5) \end{aligned}$$

当該縮小画素変換においても、上記式(5)の各係数Cub(x)は前記キュービック関数から計算される値であり、これは、求めるべき出力画素が入力画素に対し、どれだけずれているかを示す位相から計算される。上記図38に示す3:2縮小画素数変換の場合、上記Q<sub>1</sub>の出力画素の位相はその近傍の入力画素(例えばR<sub>1</sub>の入力

画素)の位相と一致しているのでその画素位相情報P<sub>1</sub>はゼロとなり、同様に上記Q<sub>1,1</sub>の出力画素の位相はその近傍の入力画素(例えばR<sub>1,1</sub>の入力画素)の位相から1/2ずれているのでその画素位相情報P<sub>1</sub>は1/2となるので、上記式(5)は式(6)のように書き換えることができる。

【0025】

$$Q_i = \text{Cub}(-1) * R_{i-1} + \text{Cub}(0) * R_i + \text{Cub}(1) * R_{i+1} + \text{Cub}(2) * R_{i+2}$$

$$Q_{i-1} = \text{Cub}(-3/2) * R_i + \text{Cub}(-1/2) * R_{i+1} + \text{Cub}(1/2) * R_{i+2} + \text{Cub}(3/2) * R_{i+3}$$

... (6)

上記Cub(x)及び入力画素の各値 $R_{i-1}$ 、 $R_i$ 、 $R_{i+1}$ 、 $R_{i+2}$ 、 $R_{i+3}$ 、...はそれぞれ既知の値であるので、この式(6)から各出力画素の補間データが計算できる。例え

$$Q_i = 0 * R_{i-1} + 1 * R_i + 0 * R_{i+1} + 0 * R_{i+2} = R_i \quad \dots (7)$$

となり、入力画素の値そのものとなる。

【0026】以上、3:2縮小画素数変換の場合を例にとって説明したが、任意の縮小比率でも同様であり、出力画素の位相さえわかれば、その位相によって前記式(1)からキュービック関数の各係数を求め、補間画素近傍の入力4点と畳み込み演算を行えばよい。

【0027】従来は、上述したような画素数変換を、例えば図39に示すようなハードワイアードな構成で実現している。なお、ここでは画像信号を輝度信号に限定して説明しており、クロマ信号についての説明は後述する。

【0028】この図39に示す構成において、直列接続されたレジスタ101~104は、それぞれ供給されたデータを1サンプル分づつ遅延するものであり、したがって、これらレジスタにより4段のシフトレジスタが構成されている。これらレジスタ101~104では、入力シフトコントロール信号IEが“H”レベルのときに、入力端子100から供給された入力画素データを順次遅延させて、それぞれ1サンプリングシフトした画像データを出力する。一方、これらレジスタ101~104において、入力シフトコントロール信号IEが“L”レベルの場合にはシフトせず前の値を保持する。上記各レジスタ101~104にてそれぞれシフトされて得られた各画像データは、それぞれ対応する乗算器111~114に送られる。

【0029】また、キュービック係数発生器105は、画素毎にキュービック係数 $C_1 \sim C_4$ を発生し、これらキュービック係数 $C_1 \sim C_4$ をそれぞれ対応する乗算器111~114に対して乗算係数として供給する。したがって、これら乗算器111~114では、上記キュービック係数発生器105で発生したキュービック係数と、上記各シフトレジスタ101~104にてそれぞれシフトされた入力画素データとをかけ算する。この乗算器11

$$Q = C_1 * D_1 + C_2 * D_2 + C_3 * D_3 + C_4 * D_4 \quad \dots (8)$$

なお、ここでは簡単のため、2:3拡大画素数変換の例を示したが、任意の拡大比率の場合は、タイミング制御が異なるだけで原理は同じであるので、それらの説明については割愛する。

【0034】また、図41には上記図39のハードウェア構成における3:2縮小画素数変換処理時の画素配置とキュービック係数 $C_1$ 、 $C_2$ 、 $C_3$ 、 $C_4$ との関係を示している。なお、図中Skipはスキップされる出力画素

ば、上記 $Q_i$ の出力画素に限って言えば、前記式(1)より、 $\text{Cub}(-1)=0$ 、 $\text{Cub}(0)=1$ 、 $\text{Cub}(1)=0$ 、 $\text{Cub}(2)=0$ なので、

1~114の乗算結果は、加算器107により加算され、FIFO(先入れ先出し)メモリ108に入力される。ただし、このメモリ108は1次元画素の記憶素子、例えばラインメモリ等でよい。

【0030】当該FIFOメモリ108は、縮小画素数変換処理の場合に画素データを飛び飛びに出力するために設けられているものであり、当該縮小画素数変換の場合にコントローラ106から供給される出力スキップ画素コントロール信号SCに基づいて飛び飛びに画素データをスキップして、出力端子109に出力する。なお、FIFOメモリ108は、拡大画素数変換処理の場合には単なるFIFOメモリとして用い、単なるディレイ素子でしかない。

【0031】コントローラ106は、拡大或いは縮小画素数変換を行う際の変換比率に基づいて、出力ポートメモリである上記FIFOメモリ108の出力スキップ画素コントロール信号SC及びシフトレジスタ101~104の入力シフトコントロール信号IEの生成、さらにキュービック係数発生器105のためのタイミングコントロールを行うものである。

【0032】図40は上記図39のハードウェア構成における2:3拡大画素数変換処理時の画素配置とキュービック係数 $C_1$ 、 $C_2$ 、 $C_3$ 、 $C_4$ との関係を示しており、当該2:3拡大画素数変換処理を行う場合にはこの図40に示すように、上記入力シフトコントロール信号IEによって3画素分入力画素データをシフトし、1画素前の画素データをシフトしないという操作を繰り返す。図39の各乗算器111~114への入力データ $D_1$ 、 $D_2$ 、 $D_3$ 、 $D_4$ は、この図40の乗算器入力 $D_1$ 、 $D_2$ 、 $D_3$ 、 $D_4$ のようになり、式(8)に示すように、これら乗算器入力とキュービック係数 $C_1$ 、 $C_2$ 、 $C_3$ 、 $C_4$ との畳み込み演算を行うことで所望の結果が得られる。

【0033】

を示している。当該縮小画素数変換処理の場合には、前記拡大画素数変換の時と異なり、上記入力シフトコントロール信号IEは常時“L”レベルとなされ、入力画素データは各レジスタ101~104にそのまま入ってくるため、各乗算器111~114の入力データ $D_1 \sim D_4$ は図41の乗算器入力 $D_1 \sim D_4$ のようになり、これとキュービック係数 $C_1 \sim C_4$ の畳み込み演算の式(8)を行うことで所望の結果が得られる。ただし、当該3:2縮

小画素数変換の場合には、出力される3画素に対して、入力の1画素が不要になるので、当該不要な画素は前記 F I F O メモリ 108 に対する書き込みをコントロールすることによってスキップする。このための制御信号が図41に示すような出力スキップ画素コントロール信号 S C となる。すなわち、この出力スキップ画素コントロール信号 S C は、“H”レベルのときスキップし、“L”レベルのときスキップしない、というように F I F O メモリ 108 を制御するための信号である。

【0035】なお、ここでは簡単のため、3:2縮小画素数変換の例を示したが任意の縮小比率の場合、そのタイミング制御が異なるだけで原理は同じであるのでここではそれらについての説明は割愛する。

【0036】次に、画素数変換と同様に、走査線数変換処理について以下にその概要を説明する。

【0037】走査線数変換処理では、先に述べた画素数変換処理の各画素を各走査線に置き換えて同様の考え方が適用できる。すなわち、走査線数変換とは、1垂直走査線期間において、入力ライン数に対して出力ライン数を所望のライン数に増減する処理であり、例えば入出力のライン数が同じであるとした場合に、ライン数を増加させたならば入力画像の垂直方向への拡大処理（拡大ライン数変換処理）となり、逆にライン数を減少させたならば入力画像の垂直方向への縮小処理（縮小ライン数変換処理）となる。

【0038】この補間方法は画素数変換と同様、様々な方法があるが、以下では、その中でも高画質化が可能なキュービック補間関数を用いた補間法について説明する。

$$\begin{aligned} Q_{1,1} &= \text{Cub}(x_{1,1}) * R_{1,1} + \text{Cub}(x_{1,2}) * R_1 + \text{Cub}(x_{1,3}) * R_{1,1} + \text{Cub}(x_{1,4}) * R_{1,2} \\ Q_{1,2} &= \text{Cub}(x_{2,1}) * R_{1,1} + \text{Cub}(x_{2,2}) * R_1 + \text{Cub}(x_{2,3}) * R_{1,1} + \text{Cub}(x_{2,4}) * R_{1,2} \\ Q_{1,3} &= \text{Cub}(x_{3,1}) * R_1 + \text{Cub}(x_{3,2}) * R_{1,2} + \text{Cub}(x_{3,3}) * R_{1,2} + \text{Cub}(x_{3,4}) * R_{1,3} \end{aligned}$$

・・・(9)

この式(9)の各係数 C u b ( x ) は前記キュービック補間関数から計算される値であり、これは、求めるべき出力ラインが入力ラインに対して、どれだけずれているかを示す位相から計算される。例えば、図42に示す2:3の拡大ライン数変換の場合、上記 Q<sub>1,1</sub> の出力ラインの位相はその近傍の入力ライン（例えば R<sub>1</sub> の入力ライン）の位相と一致しているのでそのライン位相情報 P<sub>1</sub> はゼロとなり、同様に上記 Q<sub>1,2</sub> の出力ラインの位相は

$$\begin{aligned} Q_1 &= \text{Cub}(-1) * R_{1,1} + \text{Cub}(0) * R_1 + \text{Cub}(1) * R_{1,1} + \text{Cub}(2) * R_{1,2} \\ Q_{1,1} &= \text{Cub}(-5/3) * R_{1,1} + \text{Cub}(-2/3) * R_1 + \text{Cub}(1/3) * R_{1,1} + \text{Cub}(4/3) * R_{1,2} \\ Q_{1,2} &= \text{Cub}(-4/3) * R_1 + \text{Cub}(-1/3) * R_{1,1} + \text{Cub}(2/3) * R_{1,2} + \text{Cub}(5/3) * R_{1,3} \end{aligned}$$

・・・(10)

上記 C u b ( x ) 及び入力ラインの各値 R<sub>1,1</sub>、R<sub>1</sub>、R<sub>1,1</sub>、R<sub>1,2</sub> はそれぞれ既知の値であるので、この式(10)から各出力ラインの補間データが計算できる。

$$Q_1 = 0 * R_{1,1} + 1 * R_1 + 0 * R_{1,1} + 0 * R_{1,2} = R_1$$

【0039】先ず、例えば入力ライン2本に対して、出力ライン3本を作り出すような2:3拡大ライン数変換の原理について説明する。

【0040】図42には上記2:3拡大ライン数変換の原理を説明するための図を示す。なお、この図42では、各入力ラインの値をそれぞれ R<sub>1,1</sub>、R<sub>1</sub>、R<sub>1,1</sub>、R<sub>1,2</sub>、R<sub>1,3</sub>、・・・とし、各出力ラインの値をそれぞれ Q<sub>1</sub>、Q<sub>1,1</sub>、Q<sub>1,2</sub>、Q<sub>1,3</sub>、・・・として表している。また、図42の中の P<sub>1</sub>、P<sub>2</sub>、P<sub>3</sub>、・・・は、入力ラインと出力ラインの位相のずれ（ライン位相情報）を表している。

【0041】ここで、上記3:2拡大ライン数変換においては、この図42のように入力ライン2本に対して出力ライン3本を作り出すようにしており、入力ラインと出力ラインの関係は、出力ラインの値がその近傍の入力ラインから計算されるという関係になっている。上記出力ラインを生成するための上記近傍範囲としてどのくらいまでの範囲を使用するか、或いは入力ラインから補間により出力ラインを計算する際の各係数の値としてどのような係数値を使用するかなどにより、様々な補間法が存在するが、以下の説明では、上記近傍範囲として4点（4ライン分）の範囲から補間するキュービック補間を例に挙げている。

【0042】拡大ライン数変換の場合、各出力ラインの補間値は、入力4ラインの値とキュービック関数との畳み込み演算で表され、出力ラインの補間値は以下の式(9)のように表すことができる。

【0043】

その近傍の入力ライン（例えば R<sub>1</sub> の入力ライン）の位相から 2/3 ずれているのでそのライン位相情報 P<sub>2</sub> は 2/3 となり、上記 Q<sub>1,2</sub> の出力ラインの位相はその近傍の入力ライン（例えば R<sub>1,1</sub> の入力ライン）の位相から 1/3 ずれているのでそのライン位相情報 P<sub>3</sub> は 1/3 となるので、上記式(9)は式(10)のように書き換えることができる。

【0044】

例えば、上記 Q<sub>1</sub> の出力ラインに限って言えば、前記式(1)より、Cub(-1)=0、Cub(0)=1、Cub(1)=0、Cub(2)=0なので、

・・・(11)

となり、入力ラインの値そのものとなる。

【0045】以上、2:3拡大ライン数変換の場合を例にとって説明したが、任意の拡大比率でも同様であり、出力ラインの位相さえわかれば、その位相によって式

(1) からキュービック関数の各係数を求め、補間ライン近傍の入力ライン4点と畳み込み演算を行えばよい。

【0046】次に、例えば入力ライン3個に対して出力ライン2個を作り出すような3:2縮小ライン数変換の原理について説明する。

【0047】図43には上記3:2縮小ライン数変換の原理を説明するための図を示す。なお、この図43においても前記図42と同様に、各入力ラインの値をそれぞれ $R_{1,1}$ ,  $R_1$ ,  $R_{1,1}$ ,  $R_{1,1}$ ,  $R_{1,1}$ , ...とし、各出力画素の値をそれぞれ $Q_{1,1}$ ,  $Q_{1,1}$ ,  $Q_{1,1}$ , ...と

$$Q_{1,1} = \text{Cub}(x_{1,1}) * R_{1,1} + \text{Cub}(x_{1,2}) * R_1 + \text{Cub}(x_{1,3}) * R_{1,1} + \text{Cub}(x_{1,4}) * R_{1,1}$$

$$Q_{1,1} = \text{Cub}(x_{2,1}) * R_1 + \text{Cub}(x_{2,2}) * R_{1,1} + \text{Cub}(x_{2,3}) * R_{1,1} + \text{Cub}(x_{2,4}) * R_{1,1}$$

... (12)

当該縮小ライン数変換においても、上記式(12)の各係数 $\text{Cub}(x)$ は前記キュービック関数から計算される値であり、これは、求めるべき出力ラインが入力ライン20 に対して、どれだけずれているかを示す位相から計算される。上記図43に示す3:2縮小ライン数変換の場合、上記 $Q_{1,1}$ の出力ラインの位相はその近傍の入力ライン(例えば $R_1$ の入力ライン)の位相と一致しているので

$$Q_{1,1} = \text{Cub}(-1) * R_{1,1} + \text{Cub}(0) * R_1 + \text{Cub}(1) * R_{1,1} + \text{Cub}(2) * R_{1,1}$$

$$Q_{1,1} = \text{Cub}(-3/2) * R_1 + \text{Cub}(-1/2) * R_{1,1} + \text{Cub}(1/2) * R_{1,1} + \text{Cub}(3/2) * R_{1,1}$$

... (13)

上記 $\text{Cub}(x)$ 及び入力ラインの各値 $R_{1,1}$ ,  $R_1$ ,  $R_{1,1}$ ,  $R_{1,1}$ , ...はそれぞれ既知の値であるので、この式(13)から各出力ラインの補間データが計算で 30

$$Q_{1,1} = 0 * R_{1,1} + 1 * R_1 + 0 * R_{1,1} + 0 * R_{1,1} = R_1 \quad \dots (14)$$

となり、入力ラインの値そのものとなる。

【0052】以上、3:2縮小ライン数変換の場合を例にとって説明したが、任意の縮小比率でも同様であり、出力ラインの位相さえわかれば、その位相によって前記式(1)からキュービック関数の各係数を求め、補間ライン近傍の入力4点と畳み込み演算を行えばよい。

【0053】従来は、上述したようなライン数変換を、例えば図44に示すようなハードワイアードな構成で実現している。なお、ライン数変換においては、画素数変換のように輝度信号とクロマ信号をクロマのフォーマットによって区別する必要はなく、輝度信号用とクロマ信号用は同じ回路でよい。

【0054】この図44に示す構成において、直列接続されたラインメモリ201~204は、それぞれ供給されたデータを1走査線分づつ遅延するものであり、したがって、これらにより4段のラインメモリが構成されている。これらラインメモリ201~204では、入力シフトコントロール信号IEが“H”レベルのときに、入力端子200から供給された入力ライン分の入力データ 50

して表している。また、図43の中の $P_{1,1}$ ,  $P_{1,1}$ ,  $P_{1,1}$ , ...も、入力ラインと出力ラインの位相のずれ(ライン位相情報)を表している。

【0048】ここで、上記3:2縮小ライン数変換においても、前記拡大ライン数変換同様に入力ラインと出力ラインの関係は、出力ラインの値がその近傍の入力ラインから計算されるという関係になっている。この3:2縮小ライン数変換でも、上述同様に出力ライン(補間ライン)をその近傍の入力ライン4点から補間により計算するキュービック補間を例に挙げて説明する。

【0049】すなわちこの図43の縮小ライン数変換の場合、各出力ラインの補間値(例えば $Q_{1,1}$ ,  $Q_{1,1}$ )の補間式は、以下の式(12)のようになる。

【0050】

そのライン位相情報 $P_{1,1}$ はゼロとなり、同様に上記 $Q_{1,1}$ の出力ラインの位相はその近傍の入力ライン(例えば $R_{1,1}$ の入力ライン)の位相から1/2ずれているのでそのライン位相情報 $P_{1,1}$ は1/2となるので、上記式(12)は式(13)のように書き換えることができる。

【0051】

きる。例えば、上記 $Q_{1,1}$ の出力ラインに限って言えば、前記式(1)より、 $\text{Cub}(-1)=0$ ,  $\text{Cub}(0)=1$ ,  $\text{Cub}(1)=0$ ,  $\text{Cub}(2)=0$ なので、

を順次遅延させて、それぞれ1走査線時間シフトした画像データを出力する。一方、これらラインメモリ201~204において、入力シフトコントロール信号IEが“L”レベルの場合にはシフトせずそのライン値を保持する。上記各ラインメモリ201~204にてそれぞれラインシフトされて得られた各画像データは、それぞれ対応する乗算器211~214に送られる。

【0055】また、キュービック係数発生器205は、ライン毎にキュービック係数 $C_1 \sim C_4$ を発生し、これらキュービック係数 $C_1 \sim C_4$ をそれぞれ対応する乗算器211~214に対して乗算係数として供給する。したがって、これら乗算器211~214では、上記キュービック係数発生器205で発生したキュービック係数と、上記各ラインメモリ201~204にてそれぞれラインシフトされた入力ラインデータとをかけ算する。この乗算器211~214の乗算結果は、加算器207により加算され、FIFO(先入れ先出し)フィールドメモリ208に入力される。

【0056】当該FIFOフィールドメモリ210は、



拡大ライン数変換処理の場合に必要なラインデータを飛び飛びに出力するために設けられているものであり、当該拡大ライン数変換の場合にコントローラ206から供給される入力スキップラインコントロール信号SCIに基づいてラインデータ出力する前のラインの値を保持するかを切り替え、ラインメモリ201に出力する。なお、FIFOフィールドメモリ210は、縮小ライン数変換処理の場合には単なるFIFOメモリとして用い、単なるディレイ素子でしかない。

【0057】当該FIFOフィールドメモリ208は、縮小ライン数変換処理の場合に必要なラインデータを飛び飛びに出力するために設けられているものであり、当該縮小ライン数変換の場合にコントローラ206から供給される出力スキップラインコントロール信号SCOに基づいてラインデータをスキップして、出力端子209に出力する。なお、FIFOフィールドメモリ208は、拡大ライン数変換処理の場合には単なるFIFOメモリとして用い、単なるディレイ素子でしかない。

【0058】コントローラ206は、拡大或いは縮小ライン数変換を行う際の変換比率に基づいて、出力ポート

$$Q = C_1 * D_1 + C_2 * D_2 + C_3 * D_3 + C_4 * D_4 \dots (15)$$

なお、ここでは簡単のため、2:3拡大ライン数変換の例を示したが、任意の拡大比率の場合は、タイミング制御が異なるだけで原理は同じであるので、それらの説明については割愛する。

【0061】また、図46には上記図44のハードウェア構成における3:2縮小ライン数変換処理時のライン配置とキュービック係数 $C_1, C_2, C_3, C_4$ との関係を示している。なお、図中Skipはスキップされる出力ラインを示している。当該縮小ライン数変換処理の場合には、前記拡大ライン数変換の時と異なり、上記入力シフトラインコントロール信号IEは常時“L”レベルとなされ、入力ラインデータは各ラインメモリ201~204にそのまま入ってくるため、各乗算器211~214の入力データ $D_1 \sim D_4$ は図46の乗算器入力 $D_1 \sim D_4$ のようになり、これとキュービック係数 $C_1 \sim C_4$ の畳み込み演算の式(15)を行うことで所望の結果が得られる。ただし、当該3:2縮小ライン数変換の場合には、出力される3ラインに対して、入力の1ラインが不要になるので、当該不要なラインは前記FIFOフィールドメモリ208に対する書き込みをコントロールすることによってスキップする。このための制御信号が図46に示すような出力スキップラインコントロール信号SCOとなる。すなわち、この出力スキップラインコントロール信号SCOは、“H”レベルのときスキップし、“L”レベルのときスキップしない、というようにFIFOフィールドメモリ208を制御するための信号である。

【0062】なお、ここでは簡単のため、3:2縮小ライン数変換の例を示したが任意の縮小比率の場合、その

メモリである上記FIFOフィールドメモリ208の出力スキップラインコントロール信号SCO及びラインメモリ201~204の入力シフトラインコントロール信号IEの生成、さらにキュービック係数発生器205のためのタイミングコントロールを行うものである。

【0059】図45は上記図44のハードウェア構成における2:3拡大ライン数変換処理時のライン配置とキュービック係数 $C_1, C_2, C_3, C_4$ との関係を示しており、当該2:3拡大ライン数変換処理を行う場合にはこの図45に示すように、上記入力シフトラインコントロール信号IEによって3ライン分入力ラインデータをシフトし、1ライン前のラインデータをシフトしないという操作を繰り返す。図44の各乗算器211~214への入力データ $D_1, D_2, D_3, D_4$ は、この図44の乗算器入力 $D_1, D_2, D_3, D_4$ のようになり、式(15)に示すように、これら乗算器入力とキュービック係数 $C_1, C_2, C_3, C_4$ との畳み込み演算を行うことで所望の結果が得られる。

【0060】

タイミング制御が異なるだけで原理は同じであるのでここではそれらについての説明は割愛する。

【0063】このように、画素数変換或いは走査線数変換は、従来より上述したようないわゆるASIC(Application Specific Integrated Circuit: 特定用途向けIC)等の高速積和回路を用いて実現されている。

【0064】

【発明が解決しようとする課題】しかし、上述のような様々なフォーマットに対応するため、さらには近年のように各種の新たなフォーマットが提案されてくる状況では、上記ASICの場合、その回路規模の点、或いは設計後のビット精度の変更、画素数変換アルゴリズムの変更、上記新たなフォーマットの仕様の追加等のフレキシビリティに弱いため、市場のマーケットニーズに合わせて製品化することが困難となっている。すなわち、ASICで画素数変換等を実現するには、どうしても自由度の少ない、ある固定された変換比率となるか、或いは多くても数種類程度の変換比率を切り替えて使用するというような方式に限定せざるを得ない。また、上記ASICにおいては、一度回路を作成した後は、ビット精度を変更することが容易ではなく、さらに前記VGA、SVGA、或いはXVGA、HDTV等の様々な信号フォーマットのみならず、今後出てくるであろう新たなフォーマットも含めた各種のフォーマットに全て対応させることは事実上不可能である。

【0065】また、ASICでリアルタイムに回路構成上複雑な上記フィルタスイッチング補間法で、水平、垂直の変換をかえようとするのは事実上不可能といつてよい。

【0066】特に、走査線数変換となるとどうしても外部に画像信号を蓄えるためのフィールドメモリが必要になるわけだが、このフィールドメモリから補間に必要なデータだけを読み書きを行いつつ、補間すべきラインの位相のあったものを取り出すことは、ある固定された比率以外では外部のメモリコントロールとタイミング同期をあわせることが困難である。

【0067】また、上述した説明では、入力信号のうち、輝度信号（Y信号）についての処理のみ説明したが、クロマ信号（C信号）についてはその入力フォーマットによって処理が異なる。

【0068】以下、デジタル映像信号として、例えばいわゆる4：4：4フォーマット構造、4：2：2フォーマット構造、4：1：1フォーマット構造を、それぞれ図47、図48、図49に示す。なおこれらデジタル映像信号のフォーマット構造は、既に広く知られているものであるため、ここでそれらの詳細な説明については省略する。

【0069】例えば、図47に示すいわゆるD1コンポネントフォーマット信号のような4：4：4フォーマットの場合は、2つの色差信号（R-Y（Cr）信号、B-Y（Cb）信号）が輝度信号（Y信号）と同じフォーマットであるので、これら2つの色差信号に対して前述した輝度信号の場合と全く同じ処理を施せばよい。

【0070】しかし、いわゆるD2フォーマットのような4：2：2フォーマットの場合は、図48に示すように入力クロマ信号は1サンプル毎にCrとCbが切り替わるため、輝度信号（Y信号）と同じ処理を施すことはできない。

【0071】なぜならば、4：2：2フォーマットにおけるクロマ信号の2つの色差信号CrとCbは1サンプルごとにCr、Cb、Cr、Cb、・・・と順番に繰り返すようにマルチプレックスされており、したがって、出力側でも入力時と同じ順番でCrとCbが1サンプル毎に繰り返すような構成になっていなければならない。

【0072】ところが、前述したような拡大或いは縮小画素数変換処理において、入力画素データに対するシフトレジスタでのシフトコントロール、又は出力画素データのスキップコントロール等により、画素がスキップされると、上記CrとCbの関係が画素位置により入れ違いになり、CrとCbを取り違えた画像が得られる虞れがある。この場合、CrとCbの並びを維持するためには、入力部においてCrとCbとをセットにしてスキップするようなことを行えばよい。しかし、こうすると輝度とクロマの変換後の画素数を一致させることが困難となる。

【0073】また、CrかCbのどちらか一方だけに着目して処理するようにしたとしても、入力されるCrとCbのデータ自体は1サンプルおきにしか存在しないので、例えば当該一つ飛びのデータを何らか手法を用いて

画素数変換処理したとしても（例えば、入力部でデータの一つずつスキップして詰める等する）、上記画素数変換処理後に再び輝度信号（Y信号）と画素位置が対応するように再構成しなければならない。

【0074】このため上記4：2：2フォーマットにおいて上記クロマ信号をも含めて最も簡単に画素数変換処理を行う手法としては、当該4：2：2フォーマットを一旦前記4：4：4フォーマットに変換し、出力部にて再び4：2：2フォーマットに再構成し直すことなどが存在している。

【0075】同様に、民生用の映像信号処理装置によく用いられる4：1：1フォーマットも、図49のようなフォーマット構造になっているため、演算する際にそのままクロマ信号を取り扱うことはもちろんできない。この場合は、図50に示すように4：1：1フォーマットを先ず4：2：2フォーマットに変換し、さらに4：4：4フォーマットに変換するようなことが行われる。

【0076】従って、クロマ信号が前記4：2：2フォーマットや4：1：1フォーマットとなっている場合には、一旦、前記4：4：4フォーマットにフォーマット変換を行って、輝度信号と同じフォーマット形式に変換した後、輝度信号と同じ画素数変換処理回路を用意して、画素数変換を行い、当該画素数変換処理後に、再び入力時のフォーマット形式（4：2：2フォーマットや4：1：1フォーマット）に戻すといった操作が行われる。

【0077】上述したような画素数変換を前記ASIC等のハードワイヤード構成にて実現する場合、上記4：4：4フォーマット、4：2：2フォーマット、4：1：1フォーマットという任意のクロマフォーマット全てに対応させるためには、少なくとも輝度信号用の画素数変換構成の3倍以上の回路規模が必要になるという問題がある。したがって、現実的な回路規模の制限から、上記画素数変換処理を前記ASICにて実現する場合には、ある固定された変換比率のみ可能にするとか、或いは数種類程度の変換比率のみを可能として、これらを切り替えて使用するという方式に限定されてしまい、画素数変換処理の自由度が非常に少ないものとなっている。

【0078】そこで、本発明はこのような状況に鑑みてなされたものであり、任意比率の画素数変換や走査線数変換のためのデジタル信号処理を実現可能にし、さらには、1走査線上で水平位置により変換比率が異なった画素数変換や高品位テレビジョン等にも柔軟に対応できるようにすると共に、設計後のビット精度の変更、或いは新たなフォーマットの仕様の追加等にも柔軟に対応でき、またさらに、任意のクロマフォーマットに対応した任意比率の画素数変換等をも可能にする画像信号処理装置を提供することを目的とする。

【0079】

【課題を解決するための手段】本発明は、デジタル化

された2次元画像の1次元方向の各画素に対応して配置すると共に1次元方向の各画素データが時系列に順次入力する複数の要素プロセッサと、各要素プロセッサを共通に制御するための制御手段とを備える画像信号処理装置であって、各要素プロセッサは、輝度及び色差の画素データを一時的に保存する一時保存手段と、輝度及び色差の入力画素データを格納して一時保存手段に転送する入力画素データ格納手段と、少なくとも輝度の画素の属性を表す画素属性情報を格納する画素属性情報格納手段と、輝度及び色差の画素データをスキップさせる画素スキップ情報を格納する画素スキップ情報格納手段と、画素属性情報に基づいて輝度及び色差の入力画素データ又は近傍の要素プロセッサの輝度及び色差画素データを用いた所定の演算を行う算術演算手段と、一時保存手段から取り出された輝度及び色差の入力画素データ或いは演算後の輝度及び色差の画素データを格納して出力する出力画素データ格納手段とを有してなることにより、上述した課題を解決する。また、本発明は、走査線毎に同様に処理することで、走査線数変換をも可能にしている。

【0080】ここで、本発明の画像信号処理装置では、各要素プロセッサにおいて4:4:4フォーマットの表される輝度及び色差の画素データに対してそれぞれ同一の処理を行う。また、本発明の画像信号処理装置では、各要素プロセッサの算術演算手段において、4:2:2フォーマットの輝度の画素データに対して近傍4画素の値を用いた補間演算を行い、色差（または4:1:1フォーマットを4:2:2フォーマットに変換した色差）の画素データに対して近傍画素の値を用いた直線補間演算を行う。また、本発明の画像信号処理装置では、4:2:2フォーマットの色差（または4:1:1フォーマットの色差）の画素データを、4:4:4フォーマットに変換するフォーマット変換手段を備え、各要素プロセッサにおいて4:4:4フォーマットの輝度及び色差の画素データに対して同一の処理を行う。

【0081】すなわち、本発明によれば、リアルタイムにそれぞれ独立な任意比率の画素数変換処理と走査線数変換処理とをSIMD制御のリニアアレイ型多並列プロセッサを使い、ソフトウェア処理だけで実現可能とし、変換比率はそれぞれリアルタイムに変更可能である。また、本発明によれば、例えば4:4:4フォーマットの画素データのみならず、4:2:2フォーマットや4:1:1フォーマットであっても、それぞれ独立な任意比率の画素数変換処理及び走査線数変換処理を実現可能にしている。また、外部に設けられたフィールドメモリのコントロール信号のSIMD制御のリニアアレイ型多並列プロセッサにて計算することで、外部メモリコントロール回路を不要とし、リアルタイムに変換比率を変えることを可能にする。

【0082】

【発明の実施の形態】以下、本発明の好ましい実施の形

態について、図面を参照しながら説明する。

【0083】本発明実施の形態の画像信号処理装置では、前述したハードウェア構成による画素数変換或いは走査線数変換のためのデジタル信号処理の問題点を打破するため、前記ASICのようなハードワイアード構成ではなく、DSP(digital signal processor)を用いたソフトウェアプログラムにて、当該デジタル信号処理を実現するようにしている。

【0084】このように、デジタル信号処理をソフトウェアプログラムにて行うことにより、仕様変更にも柔軟に対応でき、ソフトウェアプログラムを書き換えるだけで様々な異なる信号処理を切り替えて実行することを可能にしている。また、仕様の変更に対しても、ハードウェアは一切変更する必要がないので、いわゆるTAT(Time-Axis Transform System)期間を従来に比べかなり短縮することが可能となる。

【0085】前述した例えば画素数変換処理と走査線数変換処理とをソフトウェアプログラムにて実現するためのDSPとして、例えばいわゆるリニアアレイ型多並列プロセッサの基本的内部構成及び基本動作を、以下に説明する。

【0086】上記リニアアレイ型多並列プロセッサとは、例えば図1に示すように、入力画素の1画素に相当する要素プロセッサ40を一次元に1走査線分並べ、これら1走査線毎に並列処理することの特徴とするものである。

【0087】この図1において、入力端子30に供給された時系列の入力画素データであるシリアル入力データSIDは、各要素プロセッサ40の入力レジスタ41に入力された後、データを一時的に保存するためのローカルメモリ43に転送される。また、上記ローカルメモリ43のメモリアドレスを発生するメモリアドレスジェネレータ31とインストラクションジェネレータ32は、全ての要素プロセッサ40に対して共通な制御、すなわちいわゆるSIMD(Single Instruction Multiple Data Stream)制御を行う。

【0088】このように、リニアアレイ型多並列プロセッサの特徴でもあるが、一旦DSP内部に取り込まれたデータは、1走査線分に相当する全ての要素プロセッサ40について同じ処理が施される。具体的に言うと、各要素プロセッサ40のローカルメモリ43に転送されたデータは、それぞれ演算処理部44との間で前記補間に必要な演算が施された後、出力レジスタ42に送り込まれ、最終的にこれら各要素プロセッサ40の各出力レジスタ42から出力されることで、当該DSPからは1走査線分の補間画素データが出力画素データ（シリアル出力データSOD）として取り出されることになる。

【0089】また、各要素プロセッサ40の一つ一つは、上述したように1走査線の各画素に対応しており、各要素プロセッサ40はそれぞれ左右近傍の他の要素プ

ロセッサ 40 のローカルメモリ 43 内のデータにアクセス可能な構造になっている。このような構造を有することで、当該 DSP では、各要素プロセッサ 40 のローカルメモリ 43 に書き込まれた全体で 1 走査線分の画素データに対して、その左右近傍のデータをロードでき、これらデータをそれぞれの演算処理部 44 との間で送受して演算することにより、いわゆる水平方向の FIR フィルタ（非巡回型フィルタ）を実現できるようになっている。

【0090】また、当該 DSP では、全体として 1 走査線分の画素データを各要素プロセッサ 40 のローカルメモリ 43 に離散的に記憶でき、したがって、例えば入力時に要素プロセッサ 40 の 1 個おきに画素データを記憶することも可能である。同様に、当該 DSP では、各要素プロセッサ 40 の各ローカルメモリ 43 に記憶した全体として 1 走査線分に相当する画素データを、離散的に出力することも可能である。

【0091】これら画素数変換と走査線数変換とは当該 DSP を用いて同時に実現できるが、基本的にはそれぞれの処理はお互いに独立した処理であり、それぞれの処理を切り離して考えることができる。すなわち、信号処理としては初めに画素数変換処理を行い、次に走査線数変換処理を行うことも可能であるし、初めに走査線数変換処理を行い、次に画素数変換処理を行うことも可能である。もちろん、画素数変換処理と走査線数変換処理とを同時に行うことも可能である。以下の説明では、説明の都合上、画素数変換処理と走査線数変換処理を分けて行うこととする。まずは、輝度信号及びクロマ信号の画素数変換処理について述べる。

【0092】ここで、上述したような SIMD 制御がなされるリニアアレイ型多並列プロセッサにおいて、例えば変換比率が 2 倍以上或いは  $1/2$  倍以下等の任意の変換比率の画素数変換処理を簡易に実現する手法と、前述のハードワイヤード構成では実現不可能であった様々なフォーマットのクロマ信号での画素数変換を実現する手法とを説明する。なお、この図 2 の構成は、基本的には図 1 と同様に SIMD 制御されるものであるが、簡略化のために図 2 には主要部のみを示している。

【0093】まず、上記任意比率の画素数変換を実現する手法から説明する。

【0094】この図 2 に示す本発明の実施の形態のリニアアレイ型多並列プロセッサ 1 の各要素プロセッサ 10 は、前記同様の入力レジスタ 11 及び出力レジスタ 13 と、後述する入力スキップレジスタ 12 及び出力スキップレジスタ 14 と、ワーキングエリアを有するローカルメモリ 15 と、当該ローカルメモリ 15 又は近傍の別の要素プロセッサ 10 のローカルメモリ 15 との間でデータの送受を行って必要な演算（フィルタ演算）を行う演算処理部 16 とを主要構成要素として有してなるものである。それぞれの各要素プロセッサ 10 は入力画素の 1

画素に相当し、これら各要素プロセッサ 10 が一次元的に 1 走査線分並べられており、当該リニアアレイ型多並列プロセッサ 1 では当該 1 走査線毎の各要素プロセッサ 10 を並列処理する。

【0095】この図 2 に示すプロセッサ 1 の内部構成は、前述した図 1 と略々同じであるため、ここではそれぞれの機能についての説明は割愛するが、当該図 2 の構成は、前記図 1 の構成に対し、入力スキップレジスタ 12 と出力スキップレジスタ 14 の 2 つを明示している。以下にこれらの入力スキップレジスタ 12 及び出力スキップレジスタ 14 の動作について、他の構成要素と絡めて説明する。

【0096】この図 2 に示すリニアアレイ型多並列プロセッサ 1 において、各要素プロセッサ 10 では、入力された 1 走査線分の入力画素データを離散的或いは連続的に格納でき、また、1 走査線分に相当する画素データを離散的或いは連続的に出力することを可能にしている。

【0097】ここで、当該リニアアレイ型多並列プロセッサ 1 の各要素プロセッサ 10 において、上記離散的すなわち飛び飛びに画素データを入力或いは出力させるためには、例えば“1”でスキップ、“0”でスキップしないというような意味を持たせた画素スキップ情報を、各要素プロセッサ 10 の入力或いは出力側に割り当てれば良い。本実施の形態のプロセッサ 1 においては、このような 1 ビットからなる画素スキップ情報を上記各要素プロセッサ 10 の入力と出力側に割り当てるために、各要素プロセッサ 10 に対して上記 1 ビットの画素スキップ情報を格納する格納手段として、入力側に上記入力スキップレジスタ 12 を設け、出力側に上記出力スキップレジスタ 14 を設けるようにしている。これら入力スキップレジスタ 12、出力スキップレジスタ 14 に対して、上記画素スキップ情報を予め格納しておけば、後述するように各要素プロセッサ 10 における入力時または出力時に、画素をスキップするかスキップしないかを設定することができる。すなわち、各要素プロセッサ 10 では、入力スキップレジスタ 12、出力スキップレジスタ 14 に格納された画素スキップ情報を参照することで、入力された画素データ或いは出力する画素データをスキップするかスキップしないかを決定することができる。

【0098】より具体的に説明すると、上記入力スキップレジスタ 12 に格納された画素スキップ情報は拡大画素数変換処理時に画素データをスキップするための情報であり、当該拡大画素数変換の際に、各要素プロセッサ 10 の入力レジスタ 11 では、供給された入力画素データを上記画素スキップ情報に基づいて離散的に格納し、当該格納した入力画素データをローカルメモリ 15 に転送する。すなわち例えば、上記入力レジスタ 11 は画素スキップ情報が例えば“0”のときは入力画素データを格納し、画素スキップ情報が例えば“1”のときは入力

画素データを格納しない（スキップする、或いは後述するように値が特定されないダミーデータを格納する）。なお、当該拡大画素数変換時の出力スキップレジスタ 14 に格納される画素スキップ情報は全て“0”となり、したがって出力レジスタ 13 は画素データをそのまま出力、すなわち連続的に出力する。

【0099】一方、上記出力スキップレジスタ 14 に格納された画素スキップ情報は縮小画素数変換処理時に画素データをスキップするための情報であり、当該縮小画素数変換の際に、各要素プロセッサ 10 の出力レジスタ 13 では、上記ローカルメモリ 15 上から読み出された画素データを、当該画素スキップ情報に基づいて離散的に格納して出力する。すなわち例えば、上記出力レジスタ 13 は画素スキップ情報が例えば“0”のときは画素データを格納し、画素スキップ情報が例えば“1”のときは画素データを格納しない（スキップする）。なお、当該縮小画素数変換時の入力スキップレジスタ 12 に格納される画素スキップ情報は全て“0”となり、したがってこの場合の入力レジスタ 11 は入力画素データをそのまま、すなわち連続的にローカルメモリ 15 に転送する。

【0100】また、各要素プロセッサ 10 は、上述したようにその一つ一つが 1 走査線の各画素に対応しており、各要素プロセッサ 10 では自己のローカルメモリ 15 に保存しているデータだけでなく、それぞれ左右近傍の別の要素プロセッサ 10 のローカルメモリ 15 に保存しているデータをもアクセス可能な構造になっている。この機構により、当該リニアアレイ型多並列プロセッサ 1 においては、全要素プロセッサ 10 が同時にその左右近傍の他の要素プロセッサ 10 のローカルメモリ 15 に格納されているデータをロードでき、したがって、各要素プロセッサ 10 の演算処理部 16 ではこれらデータを使用して後述するような例えば FIR フィルタ演算を実現可能になっている。上記演算処理部 16 でのフィルタ演算により生成された画素データは、再び自己のローカルメモリ 15 内に格納される。

【0101】ただし、リニアアレイ型多並列プロセッサの特徴でもある SIMD 制御により、1 走査線分に相当する全ての要素プロセッサ 10 では、同じ処理が行われる。すなわち、図 2 では図示を省略しているが、上記演算処理部 16 の命令コードを生成する前記インストラクションジェネレータ 32 と、ローカルメモリ 15 のアドレスデータを生成するメモリアドレスジェネレータ 31 は、全ての要素プロセッサ 10 に対して共通な制御を行う。なお、上記左右近傍の他の要素プロセッサ 10 へのアクセスも、上記 SIMD 制御より全要素プロセッサ 10 共通動作であり、同時に各要素プロセッサ 10 毎に異なるアクセスはできない。

【0102】上記ローカルメモリ 15 に格納された上記画素データは、出力レジスタ 13 に送り込まれ、最終的

に出力レジスタ 13 からシリアル出力データ SOD として 1 走査線分のデータが出力されることになる。

【0103】縮小画素数変換の場合には、出力スキップレジスタ 14 に格納されている画素スキップ情報に基づいて、上記ローカルメモリ 15 から読み出された画素データが出力レジスタ 13 に離散的に格納され、この格納された画素データが出力レジスタ 13 から出力されることになる。

【0104】次に、前述したキュービック補間関数を使った画素数変換を、上記図 2 に示したりニアアレイ型多並列プロセッサ 1 にて実現する方法について説明する。なお、画素数変換では、画素数の拡大と縮小の他に等倍（すなわち 1 : 1）変換もあるが、これは拡大、縮小の境界条件であり、どちらかに含めることができるので、ここでは等倍変換については拡大に含めることにする。

【0105】先ず、拡大画素数変換の例から説明する。

【0106】拡大画素数変換では、前記図 40 にて説明したように、画素データの入力時に変換比率（拡大比率）に応じて入力画素データを飛び飛びに配置する必要がある。

【0107】本実施の形態では、当該拡大画素数変換時に入力画素データを飛び飛びに配置（出力画素は連続的に配置）するための情報として、上記画素スキップ情報を使用されている。また、当該拡大画素数変換では、出力画素データを、入力時にスキップされた画素を除く所望の出力画素近傍の 4 点の入力画素データと、それぞれの画素に対応する位相情報から計算される各 4 点のキュービック係数との畳み込み演算を行うことで生成しなければならない。このように、拡大画素数変換を行うためには、上記画素スキップ情報と、キュービック係数を計算するための各画素に対応する位相情報と、所望の出力画素を生成するための当該出力画素近傍の 4 点の画素データとが、必要となる。

【0108】図 3 には、拡大画素数変換処理の全体の流れを示す。なお、ここでは上記拡大画素数変換として、L : K に拡大する例を挙げている。ただし、K 及び L は正の整数であり、 $K \geq L$  である。K = L の場合は等倍変換（1 : 1）となる。

【0109】この図 3 において、先ず、ステップ S T 1 では上記 L : K の変換比率（拡大比率）が設定される。

【0110】次のステップ S T 2 では、全要素プロセッサ 10 において、画素スキップ情報と位相情報を計算する。なお、当該ステップ S T 2 における画素スキップ情報と位相情報の計算の詳細については後述する。

【0111】ステップ S T 3 では、次ラインの画素データの入力が可能か否かの判断を行い、当該次ラインの画素データの入力が可能となるまでこの判断を繰り返す。当該ステップ S T 3 にて次ラインの画素データの入力が可能になると、次のステップ S T 4 では、1 ライン分の画素データの入力となされる。

【0112】ステップST5では、自己の要素プロセッサ10への入力画素データと当該自己の要素プロセッサ10の近傍の4つの要素プロセッサ10の画素データ（近傍4点の画素データ）とを、自己の要素プロセッサ10内のローカルメモリ15に格納する。なお、当該ステップST5における処理の詳細については後述する。

【0113】ステップST6では位相情報からキュービック係数を計算する。すなわち、前述した式(1)を用いた計算を行う。

【0114】ステップST7では、上記ステップST6にて求めたキュービック係数と上記近傍の4点の画素データの畳み込み演算を行う。なお、この畳み込み演算の詳細については後述する式(16)にて説明する。

【0115】ステップST8では、1ライン分の演算を行い、得られた補間画素データを出力する。その後はステップST3以降の処理に戻る。

【0116】次に、上述のステップST2における画素スキップ情報と位相情報は、当該リニアアレイ型多並列プロセッサ(DSP)外部の例えばCPU(中央処理装置)等のプリプロセッサにて予め計算しておき、実際の画素データとの畳み込み演算を行う前に入力レジスタ11を介して入力しておくようなことが可能である。上記画素スキップ情報は入力スキップレジスタ12と出力スキップレジスタ14に格納され、上記位相情報は例えば上記ローカルメモリ15内に設けた位相情報格納レジスタに保存されることになる。

【0117】一方、上記画素スキップ情報と位相情報は、SIMD制御されたリニアアレイ型多並列プロセッサ1内部で、例えば映像ブランピング区間や電源投入時等に計算することも可能である。すなわち、例えば水平方向の画素の位置によって変換比率が変わらないような1走査線中で変換比率が一定である画素数変換の場合には、上記画素スキップ情報と位相情報は当該プロセッサ(DSP)1内で容易に計算できる。このように画素スキップ情報と位相情報の生成も含む全ての処理を当該プロセッサ1内部だけで実現するようにすれば、全体のシステム構成をシンプルにすることができる。なお、この場合も上記画素スキップ情報は入力スキップレジスタ12と出力スキップレジスタ14に格納され、上記位相情報はローカルメモリ15内の位相情報格納レジスタに保存される。

【0118】以下に、拡大画素数変換時において、上述したようなSIMD制御された当該プロセッサ1内部で画素スキップ情報と位相情報を計算する方法について説明する。なお、画素スキップ情報の計算と位相情報の計算は、必ずしも同時に行う必要はないが、当該画素スキップ情報の設定と位相情報の計算においては共通した演算が多く、当該プロセッサ1のローカルメモリ15内のレジスタのリソースとプログラムメモリアリア、実行ステップ数等の点から共有化した方が効率がよいので、こ

こでは画素スキップ情報と位相情報を同時に求めるアルゴリズム例を示す。この計算は前述したように映像ブランピング区間、あるいは電源投入後に行えばよい。

【0119】図4には、SIMD制御されたリニアアレイ型多並列プロセッサ1内部で上記画素スキップ情報と位相情報を計算する手順を示す。

【0120】前記図3のステップST1の次に進む前記ステップST2では、この図4に示すステップST11の以降の処理が行われる。

【0121】先ず、ステップST11では、ローカルメモリ15に位相情報を保存するための領域(前記位相情報格納レジスタ $d_{11}$ )を確保する。

【0122】ステップST12では、全要素プロセッサ10において、それぞれ1つ左側の要素プロセッサ10のローカルメモリ15の位相情報格納レジスタ $d_{11}$ に保存されている値(位相情報)にLを加算し、得られた値を自己のローカルメモリ15の位相情報格納レジスタ $d_{11}$ に格納する。

【0123】次のステップST13では、上記ステップST12で加算されて位相情報格納レジスタ $d_{11}$ に格納された値がKの値以上であるか否かの判定を行う。このステップST13にてKの値以上であると判定したときにはステップST14に進み、Kの値未満であると判定したときにはステップST16に進む。

【0124】ステップST14では、上記位相情報格納レジスタ $d_{11}$ に格納された値からKを引いて、得られた値を再び当該位相情報格納レジスタ $d_{11}$ に格納する。次のステップST15では、入力スキップレジスタ12に"0"を格納する。すなわち、当該要素プロセッサ10に入力された画素データはスキップせずにそのまま入力されるようにする。

【0125】一方、上記ステップST13において位相情報格納レジスタ $d_{11}$ に格納された値が上記Kの値未満であると判定したときに進むステップST16では、入力スキップレジスタ11に"1"を格納する。すなわち、当該要素プロセッサ10に入力された画素データが一つスキップして入力されるようにする。

【0126】上記ステップST15及びステップST16の後は、ステップST17に進む。このステップST17では、1ライン分の全画素(少なくとも全要素プロセッサ10の数より多い)について上述の処理が終了したか否かの判定を行い、終了していないと判断したときにはステップST12に戻って上述の処理を繰り返し、終了したと判断したときには次の処理すなわち前記図3のステップST3の処理に進む。

【0127】ただし、この図4に示したアルゴリズムにおいては、各要素プロセッサ10はそれぞれ左側の要素プロセッサ10の位相情報格納レジスタ $d_{11}$ に格納された値を自己の要素プロセッサ10内部に格納する操作を行うが、最左端の要素プロセッサ10はその左の要素プ

ロセッサが存在しないためその値は確定しないことになる。したがって、当該最左端の要素プロセッサ 10 の場合は、常に“0”を位相情報格納レジスタ  $d_{11}$  に格納するものとする。

【0128】次に、前記図 3 のフローチャートのステップ ST 5 における処理、すなわち、各要素プロセッサ 10 において、近傍の 4 つの要素プロセッサから画素データ（近傍 4 点の画素データ）を自己の要素プロセッサ 10 内に格納する手順を、図 5 及び図 6 のフローチャートを用いて説明する。

【0129】前記図 3 のステップ ST 4 の次に進む前記ステップ ST 5 では、図 5 に示すステップ ST 21 の以降の処理が行われる。なおこのとき、前記図 4 のフローチャートに示した操作の結果、各要素プロセッサ 10 のローカルメモリ 15 には、上記入力スキップレジスタ 12 に格納された画素スキップ情報に基づいてスキップ入力された入力画素データと各画素に対応する位相情報が格納されている。

【0130】まず、ステップ ST 21 では、全要素プロセッサ 10 において、それぞれ内部のローカルメモリ 15 上に、それぞれ近傍の 4 つの要素プロセッサ 10 にて入力時にスキップされた画素を除く 4 つの近傍画素データを格納するための領域（レジスタ  $d_{11}$ 、 $d_{12}$ 、 $d_{13}$ 、 $d_{14}$  とする）を確保すると共に、1 ビットのレジスタ  $d_{15}$  を確保する。なお、変換比率（拡大比率）を例えば 1 倍から 2 倍までとすると、上記レジスタ  $d_{11}$  には自己の要素プロセッサ 10 の 1 つ左隣の要素プロセッサ 10 からの画素データが格納され、レジスタ  $d_{12}$  には自己の要素プロセッサ 10 の画素データが、レジスタ  $d_{13}$  には自己の要素プロセッサ 10 の 1 つ右隣の要素プロセッサ 10 からの画素データが、レジスタ  $d_{14}$  には自己の要素プロセッサ 10 の 2 つ右隣の要素プロセッサからの画素データが格納される。また、上記レジスタ  $d_{15}$  は画素スキップ情報が格納されるため、以下、スキップ情報格納レジスタ  $d_{15}$  と呼ぶことにする。

【0131】次のステップ ST 22 では、全要素プロセッサ 10 において、入力スキップレジスタ 11 に格納された画素スキップ情報が“1”であるか否かの判定を行い、“1”であると判定した場合にはステップ ST 23 以降に進み、“1”でない（“0”である）と判定した場合にはステップ ST 25 以降に進む。

【0132】ステップ ST 22 にて入力スキップレジスタ 11 に格納された画素スキップ情報が“1”であると判定された場合、ステップ ST 23 では全要素プロセッサ 10 においてそれぞれ 1 つ左隣の要素プロセッサ 10 の入力画素データを自己の要素プロセッサ 10 のレジスタ  $d_{11}$  に格納する。次のステップ ST 24 では全要素プロセッサ 10 においてそれぞれ 1 つ左隣の要素プロセッサ 10 の入力スキップレジスタ 11 の値を、自己の要素プロセッサ 10 のローカルメモリ 15 内の上記スキップ

情報格納レジスタ  $d_{15}$  に格納する。

【0133】一方、ステップ ST 22 にて入力スキップレジスタ 11 に格納された画素スキップ情報が“1”でない（“0”である）と判定された場合、ステップ ST 25 では全要素プロセッサ 10 においてそれぞれ自己の要素プロセッサ 10 の入力画素データの値を自己の要素プロセッサ 10 の上記レジスタ  $d_{11}$  に格納し、次のステップ ST 26 では自己の要素プロセッサ 10 の入力スキップレジスタ 11 の値を自己の要素プロセッサ 10 のローカルメモリ 15 内の上記スキップ情報格納レジスタ  $d_{15}$  に格納する。

【0134】次に、ステップ ST 27 では、全要素プロセッサ 10 においてそれぞれ 1 つ左隣の要素プロセッサ 10 のローカルメモリ 15 上の上記レジスタ  $d_{11}$  の値（ビット）が“1”であるか否かの判定を行い、“1”であると判定した場合にはステップ ST 28 以降に進み、“1”でない（“0”である）と判定した場合にはステップ ST 29 以降に進む。

【0135】ステップ ST 27 にて“1”であると判定された場合、ステップ ST 28 では、全要素プロセッサ 10 においてそれぞれ自己の要素プロセッサ 10 の 2 つ左隣の要素プロセッサ 10 のローカルメモリ 15 上の画素データを、自己のローカルメモリ 15 の上記レジスタ  $d_{11}$  に格納する。

【0136】一方、ステップ ST 27 にて“1”でないと判定された場合、ステップ ST 29 では、全要素プロセッサ 10 においてそれぞれ自己の要素プロセッサ 10 の 1 つ左隣の要素プロセッサ 10 のローカルメモリ 15 上の画素データを、自己のローカルメモリ 15 の上記レジスタ  $d_{11}$  に格納する。

【0137】次に処理は図 6 のフローチャートのステップ ST 31 に進み、このステップ ST 31 では、全要素プロセッサ 10 においてそれぞれ 1 つ右隣の要素プロセッサ 10 のローカルメモリ 15 上の上記スキップ情報格納レジスタ  $d_{15}$  の値（ビット）が“1”であるか否かの判定を行い、“1”であると判定した場合にはステップ ST 32 以降に進み、“1”でない（“0”である）と判定した場合にはステップ ST 34 以降に進む。

【0138】ステップ ST 31 にて“1”であると判定された場合、ステップ ST 32 では、全要素プロセッサ 10 においてそれぞれ自己の要素プロセッサ 10 の 2 つ右隣の要素プロセッサ 10 のローカルメモリ 15 上の画素データを、自己のローカルメモリ 15 の上記レジスタ  $d_{11}$  に格納し、ステップ ST 33 では、上記レジスタ  $d_{15}$  の値を 2 つ分左にシフト（左側の 2 つの要素プロセッサ 10 のレジスタ  $d_{15}$  に順次シフト）する。

【0139】一方、ステップ ST 31 にて“1”でないと判定された場合、ステップ ST 34 では、全要素プロセッサ 10 においてそれぞれ自己の要素プロセッサ 10 の 1 つ右隣の要素プロセッサ 10 のローカルメモリ 15



上の画素データを、自己のローカルメモリ 15 の上記レジスタ  $d_{11}$  に格納し、ステップ ST 35 では、上記レジスタ  $d_{11}$  の値を 1 つ分左にシフト（左側の 1 つの要素プロセッサ 10 のレジスタ  $d_{11}$  にシフト）する。

【0140】次のステップ ST 36 では、全要素プロセッサ 10 においてそれぞれ 1 つ右隣の要素プロセッサ 10 のローカルメモリ 15 上の上記レジスタ  $d_{11}$  の値（ビット）が“1”であるか否かの判定を行い、“1”であると判定した場合にはステップ ST 37 以降に進み、“1”でない（“0”である）と判定した場合にはステップ ST 38 以降に進む。

【0141】ステップ ST 36 にて“1”であると判定された場合、ステップ ST 37 では、全要素プロセッサ 10 においてそれぞれ自己の要素プロセッサ 10 の 2 つ右隣の要素プロセッサ 10 のローカルメモリ 15 上の画素データを、自己のローカルメモリ 15 の上記レジスタ  $d_{11}$  に格納する。

【0142】一方、ステップ ST 36 にて“1”でないと判定された場合、ステップ ST 38 では、全要素プロセッサ 10 においてそれぞれ自己の要素プロセッサ 10 の 1 つ右隣の要素プロセッサ 10 のローカルメモリ 15 上の画素データを、自己のローカルメモリ 15 の上記レジスタ  $d_{11}$  に格納する。

【0143】その後は、次の処理すなわち前記図 3 のステップ ST 6 の処理に進む。

【0144】以上の処理は、リニアアレイ型多並列プロセッサ 1 においては、例えばフラグによってデータを移動する／しないという処理の繰り返しを指示するだけなので非常に僅かなステップ数で演算は終了する。なお、拡大比率を 1 倍から 2 倍までに限定したのは考え方を単純にするためであり、2 倍以上の場合は、近傍の要素プロセッサ間で画素データを通信する範囲が広がるだけである。この場合も基本的な考え方は同じなので説明は省略する。

【0145】上述した前記図 3 のステップ ST 2（図 4 のフローチャート）と、図 3 のステップ ST 5（図 5 及び図 6 のフローチャート）との操作を行うことで、上記ローカルメモリ 15 のレジスタ  $d_{11}$ 、 $d_{12}$ 、 $d_{13}$ 、 $d_{14}$  に格納される 4 つの画素データと、同じくローカルメモリ 15 の前記位相情報格納レジスタ  $d_{15}$  に格納される各

$$Q = C_1((K+Ph)/K) \cdot d_{11} + C_2(Ph/K) \cdot d_{12} + C_3((K-Ph)/K) \cdot d_{13} + C_4((2K-Ph)/K) \cdot d_{14}$$

ただし、 $C_1(x) = -|x|^3 + 5|x|^2 - 8|x| + 4$   
 $C_2(x) = |x|^3 - 2|x|^2 + 1$

次に、縮小画素数変換の場合について説明する。

【0150】縮小画素数変換は、前述した拡大画素数変換とは逆に、画素データの入力時に変換比率（縮小比率）に応じて入力画素データが飛び飛びに配置されるこ

画素に対応する位相情報とは、図 7 に示すような関係となる。なお、この図 7 には入力スキップレジスタ 12 及び出力スキップレジスタ 14 にそれぞれ格納される画素スキップ情報も示している。また、この図 7 の例では、一つの要素プロセッサ 10 が当該図 7 の縦方向の 1 列と対応しており、図中 Y 及び Q はそれぞれ画素データを示し、Ph は位相情報を示している。この図 7 において、拡大画素数変換の場合、画素スキップ情報にてスキップされた部分にはダミーデータ M が入れられ、入力の段階でトータルの画素数を出力の画素数と等しくしている。なお、上記ダミーデータ M は、入力時に予め画素数を増やし、SIMD 制御に適したデータ配列にするためのもので、実際の畳み込み演算ではこのダミーデータ M は使われない。したがって、このダミーデータ M の値は何であって構わない。このダミーデータ M を挿入するかしないか、言い換えれば入力画素データのスキップを行うか否かの設定は、1 ビットあれば充分である。すなわち例えば、“1”でダミーデータ M を入れる（入力画素データはスキップする）、“0”でダミーデータ M を入れない（入力画素データはスキップしない）ことにすればよい。このようなダミーデータ M を入れるか否かを示す情報が前記画素スキップ情報である。

【0146】次に、前記図 3 のフローチャートのステップ ST 7 における処理、すなわち、拡大画素数変換時のキュービック係数と上記近傍の 4 点の画素データの畳み込み演算について説明する。

【0147】上述のようにしてローカルメモリ 15 上の各レジスタ  $d_{11}$ 、 $d_{12}$ 、 $d_{13}$ 、 $d_{14}$  に格納された 4 つの近傍画素のデータと位相情報格納レジスタ  $d_{15}$  に格納された位相情報とが得られた後は、これら各画素に対応するキュービック係数を求め、畳み込み演算を行う。

【0148】上記キュービック係数は前記式（1）から計算されるが、実際は  $|x|$  の大きさによって 2 つの場合に分けられ、最終的な出力は以下の式（16）で与えられる。なお、拡大画素数変換の場合、出力スキップレジスタ 14 の画素スキップ情報は全て“0”に設定され、出力画素データのスキップは行われないので、以下の式（9）による値がそのまま出力画素データとなる。

【0149】

$$\dots (16)$$

とはないが、前述の図 4 1 にて説明したように、画素データの出力時に変換比率（縮小比率）に応じて出力画素データを飛び飛びにスキップさせながら出力することが行われる。本実施の形態では、当該縮小画素数変換時に出力画素データを飛び飛びに配置（入力画素は連続的に配置）するための情報として、上記画素スキップ情報が



使用されている。また、当該縮小画素数変換の場合の補間画素データは、前記拡大時と同様に、所望の出力画素近傍の4点の入力画素データと、それぞれの画素に対応する位相情報から計算される各4点のキュービック係数との畳み込み演算を行うことで生成される。このような縮小画素数変換を行うためには、上記画素スキップ情報と、キュービック係数を計算するための各画素に対応する位相情報と、所望の出力画素を生成するための上記近傍4点の画素データとが、必要となる。

【0151】図8には、縮小画素数変換処理の全体の流れを示す。なお、ここでは上記縮小画素数変換として、 $L:K$ に縮小する例を挙げている。ただし、 $K$ 及び $L$ は正の整数であり、 $K < L$ である。

【0152】この図8において、まず、ステップST41では上記 $L:K$ の変換比率（拡大比率）が設定される。

【0153】次のステップST42では、全要素プロセッサ10において、画素スキップ情報と位相情報を計算する。なお、当該ステップST42における画素スキップ情報と位相情報の計算の詳細については後述する。

【0154】ステップST43では、次ラインの画素データの入力が可能か否かの判断を行い、当該次ラインの画素データの入力が可能となるまでこの判断を繰り返す。当該ステップST43にて次ラインの画素データの入力が可能になると、次のステップST44では、1ライン分の画素データの入力となされる。

【0155】ステップST45では、自己の要素プロセッサ10への入力画素データと当該自己の要素プロセッサ10の近傍の4つの要素プロセッサ10の画素データ（近傍4点の画素データ）とを、自己の要素プロセッサ10内のローカルメモリ15に格納する。なお、当該ステップST45における処理の詳細については後述する。

【0156】ステップST46では位相情報からキュービック係数を計算する。すなわち、前述した式(1)を用いた計算を行う。

【0157】ステップST47では、上記ステップST46にて求めたキュービック係数と上記近傍の4点の画素データの畳み込み演算を行う。なお、この畳み込み演算の詳細については後述する式(17)にて説明する。

【0158】ステップST48では、1ライン分の演算を行い、得られた補間画素データを出力する。その後はステップST43以降の処理に戻る。

【0159】この縮小画素数変換の場合も、前記拡大画素数変換時と同様に、上述のステップST42における画素スキップ情報と位相情報は、当該リニアアレイ型多並列プロセッサ(DSP)外部の例えばCPU(中央処理装置)等のプリプロセッサにて予め計算しておき、実際の画素データとの畳み込み演算を行う前に入力レジスタ11を介して入力しておくようなことが可能である。

上記画素スキップ情報は入力スキップレジスタ12と出力スキップレジスタ14に格納され、上記位相情報は例えば上記ローカルメモリ15内に設けた位相情報格納レジスタに保存されることになる。また、上記画素スキップ情報と位相情報は、前記拡大画素数変換時と同様に、SIMD制御されたリニアアレイ型多並列プロセッサ1内部で、例えば映像ブランキング区間や電源投入時等に計算することも可能である。すなわち、例えば水平方向の画素の位置によって変換比率が変わらないような1走査線中で変換比率が一定である画素数変換の場合には、上記画素スキップ情報と位相情報は当該プロセッサ(DSP)1内で容易に計算できる。このように画素スキップ情報と位相情報の生成も含む全ての処理を当該プロセッサ1内部だけで実現するようにすれば、全体のシステム構成をシンプルにすることができる。なお、この場合も上記画素スキップ情報は入力スキップレジスタ12と出力スキップレジスタ14に格納され、上記位相情報はローカルメモリ15内の位相情報格納レジスタに保存される。

【0160】以下に、縮小画素数変換時において、上述したようなSIMD制御された当該プロセッサ1内部で画素スキップ情報と位相情報を計算する方法について説明する。なお、この縮小画素数変換の場合も、画素スキップ情報の計算と位相情報の計算は、必ずしも同時に行う必要はないが、当該画素スキップ情報の設定と位相情報の計算においては共通した演算が多く、当該プロセッサ1のローカルメモリ15内のレジスタのリソースとプログラムメモリエリア、実行ステップ数等の点から共有化した方が効率がよいので、ここでは画素スキップ情報と位相情報を同時に求めるアルゴリズム例を示す。この計算は前述したように映像ブランキング区間、あるいは電源投入後に行えばよい。

【0161】図9には、SIMD制御されたリニアアレイ型多並列プロセッサ1内部で上記縮小画素数変換における画素スキップ情報と位相情報を計算する手順を示す。

【0162】前記図8のステップST41の次に進む前記ステップST2では、この図9に示すステップST51の以降の処理が行われる。

【0163】まず、ステップST51では、ローカルメモリ15に位相情報を保存するための領域（前記位相情報格納レジスタd<sub>11</sub>）とワーキングレジスタを確保する。

【0164】ステップST52では、全要素プロセッサ10において、それぞれ1つ左側の要素プロセッサ10のローカルメモリ15の位相情報格納レジスタd<sub>11</sub>に保存されている値（位相情報）に $L$ を加算し、得られた値を自己のローカルメモリ15のワーキングレジスタに格納する。

【0165】次のステップST53では、全要素プロセ

ッサ 10 においてそれぞれ自己の要素プロセッサ 10 の 1 つ左隣の要素プロセッサ 10 の位相情報格納レジスタ  $d_{i-1}$  に格納された値から  $K$  を減算した値を、自己の要素プロセッサ 10 のローカルメモリ 15 内の位相情報格納レジスタ  $d_i$  に格納する。

【0166】ステップ ST 54 では、上記ステップ ST 52 で加算されてワーキングレジスタに格納された値が  $K$  の 2 倍の値より小さいか否かの判定を行う。このステップ ST 54 にて  $K$  の 2 倍の値より小さいと判定したときにはステップ ST 55 に進み、 $K$  の 2 倍の値以上である

と判定したときにはステップ ST 57 に進む。  
【0167】ステップ ST 55 では、上記位相情報格納レジスタ  $d_{i-1}$  に格納された値に  $L$  を加算して、得られた値を再び当該位相情報格納レジスタ  $d_{i-1}$  に格納する。次のステップ ST 56 では、出力スキップレジスタ 14 に "0" を格納する。すなわち、当該要素プロセッサ 10 のローカルメモリ 15 から取り出された画素データがスキップせずにそのまま出力レジスタ 13 に格納して出力されるようにする。

【0168】一方、上記ステップ ST 54 においてワーキングレジスタに格納された値が上記  $K$  の 2 倍の値以上であると判定したときに進むステップ ST 57 では、出力スキップレジスタ 14 に "1" を格納する。すなわち、当該要素プロセッサ 10 のローカルメモリ 15 から取り出された画素データが 1 つスキップして出力レジスタ 13 から出力されるようにする。

【0169】上記ステップ ST 56 及びステップ ST 57 の後は、ステップ ST 58 に進む。このステップ ST 58 では、1 ライン分の全画素（少なくとも全要素プロセッサ 10 の数より多い）について上述の処理が終了したか否かの判定を行い、終了していないと判断したときにはステップ ST 52 に戻って上述の処理を繰り返し、終了したと判断したときには次の処理すなわち前記図 8 のステップ ST 43 の処理に進む。

【0170】ただし、この図 9 に示したアルゴリズムにおいても前述した拡大画素数変換時と同様に、各要素プロセッサ 10 はそれぞれ左側の要素プロセッサ 10 の位相情報格納レジスタ  $d_{i-1}$  に格納された値を自己の要素プロセッサ 10 内部に格納する操作を行うが、最左端の要素プロセッサ 10 はその左の要素プロセッサが存在しないためその値は確定しないことになる。したがって、当該最左端の要素プロセッサ 10 の場合は、常に "0" を位相情報格納レジスタ  $d_{i-1}$  に格納するものとする。

【0171】次に、前記図 8 のフローチャートのステップ ST 45 における処理、すなわち、各要素プロセッサ 10 において、近傍の 4 つの要素プロセッサから画素データ（近傍 4 点の画素データ）を自己の要素プロセッサ 10 内に格納する手順を、図 10 のフローチャートを用いて説明する。

【0172】前記図 8 のステップ ST 44 の次に進む前

記ステップ ST 45 では、図 10 に示すステップ ST 61 の以降の処理が行われる。なおこのとき、前記図 8 のフローチャートに示した操作の結果、各要素プロセッサ 10 のローカルメモリ 15 には、入力画素データと各画素に対応する位相情報が格納されている。

【0173】まず、ステップ ST 61 では、全要素プロセッサ 10 において、それぞれ内部のローカルメモリ 15 上に、それぞれ近傍の 4 つの要素プロセッサ 10 にて入力時にスキップされた画素を除く 4 つの近傍画素データを格納するための領域（レジスタ  $d_{i-1}$ ,  $d_i$ ,  $d_{i+1}$ ,  $d_{i+2}$  とする）を確保すると共に、1 ビットのスキップ情報格納レジスタ  $d_{i-2}$  を確保する。なお、変換比率（拡大比率）を例えば 1 倍から 2 倍までとすると、これらレジスタ  $d_{i-1}$  には自己の要素プロセッサ 10 の 1 つ左隣の要素プロセッサ 10 からの画素データが格納され、レジスタ  $d_i$  には自己の要素プロセッサ 10 の画素データが、レジスタ  $d_{i+1}$  には自己の要素プロセッサ 10 の 1 つ右隣の要素プロセッサ 10 からの画素データが、レジスタ  $d_{i+2}$  には自己の要素プロセッサ 10 の 2 つ右隣の要素プロセッサからの画素データが格納される。これは前述した拡大画素数変換時と同じである。

【0174】次のステップ ST 62 では、全要素プロセッサ 10 において、入力画素データの値をローカルメモリ 15 のレジスタ  $d_i$  に格納する。

【0175】次のステップ ST 63 では、全要素プロセッサ 10 において、それぞれ自己の要素プロセッサ 10 の一つ左隣の要素プロセッサ 10 の前記レジスタ  $d_{i-1}$  に格納されている入力画素データを、自己の要素プロセッサ 10 のレジスタ  $d_{i-1}$  に格納する。

【0176】ステップ ST 64 では、全要素プロセッサ 10 において、それぞれ自己の要素プロセッサ 10 の一つ右隣の要素プロセッサ 10 の前記レジスタ  $d_{i+1}$  に格納されている入力画素データを、自己の要素プロセッサ 10 のレジスタ  $d_{i+1}$  に格納する。

【0177】次のステップ ST 65 では、全要素プロセッサ 10 において、それぞれ自己の要素プロセッサ 10 の一つ右隣の要素プロセッサ 10 の前記レジスタ  $d_{i+1}$  に格納されている入力画素データを、自己の要素プロセッサ 10 のレジスタ  $d_{i+1}$  に格納する。

【0178】その後は、図 8 のフローチャートのステップ ST 46 に進むことになる。

【0179】以上の処理は、リニアアレイ型多並列プロセッサ 1 において各要素プロセッサ 10 のそれぞれ近傍の画素データの通信機能を使用するだけの非常に僅かなステップ数で演算は終了する。

【0180】上述した前記図 8～図 10 の操作を行うことで、上記ローカルメモリ 15 のレジスタ  $d_{i-1}$ ,  $d_i$ ,  $d_{i+1}$ ,  $d_{i+2}$  に格納される 4 つの画素データと、同じくローカルメモリ 15 の前記位相情報格納レジスタ  $d_{i-2}$  に格納される各画素に対応する位相情報とは、図 11 に示す

ような関係となる。なお、この図 11 は、前記図 7 と同様に表しており、図 11 の図中 G s がスキップされた出力画素データを表している。

【0181】次に、前記図 8 のフローチャートのステップ ST 47 における処理、すなわち、縮小画素数変換時のキュービック係数と上記近傍の 4 点の画素データの畳み込み演算について説明する。

【0182】上述のようにしてローカルメモリ 15 上の各レジスタ  $d_{11}$ ,  $d_{12}$ ,  $d_{13}$ ,  $d_{14}$  に格納された 4 つの近傍画素のデータと位相情報格納レジスタ  $d_{15}$  に格納さ

$$Q = C_1((K+Ph)/K) * d_{11} + C_2(Ph/k) * d_{12} + C_3((k-Ph)/K) * d_{13} + C_4((2K-Ph)/K) * d_{14}$$

... (17)

ただし、 $C_1(x) = -|x|^3 + 5|x|^2 - 8|x| + 4$   
 $C_2(x) = |x|^3 - 2|x|^2 + 1$

以上のように SIMD 制御のリニアアレイ型多並列プロセッサ 1 では、前述した各式に沿って高々一回の計算で全画素に対する出力画素データを同時に計算できる。このように、本発明実施の形態のリニアアレイ型多並列プロセッサ 1 では、SIMD 制御であることを利用することにより、少ないステップ数で、画素数変換処理が可能

である。  
 【0185】上述した説明は、全て輝度信号を例に挙げており、以下に SIMD 制御のリニアアレイ型多並列プロセッサ 1 においてクロマ信号の画素数変換を行う場合の具体例を説明する。

【0186】先ず、前述した 4:4:4 フォーマットにおけるクロマ信号の画素数変換では、クロマ信号のフォーマットが輝度信号のフォーマットと同じであるので、前述した輝度信号の画素数変換と同様の処理でよい。リニアアレイ型多並列プロセッサ 1 では、ハードウェアは

なんら追加する必要がなく、クロマ信号用のソフトウェアを追加するだけで済むため、新たに回路が増えることはない。

【0187】次に、前述した 4:2:2 フォーマットにおけるクロマ信号の画素数変換について説明する。

【0188】当該 4:2:2 フォーマットでは、前述の図 48 に示したように、画素毎に Cr と Cb が交互に繰り返り並んでおり、各色差信号は 1 個おきにしか存在しない。また、前述したように、輝度と色の画素位置と画素数を合わせるためには、輝度信号と同様に、画素データの入力時に Cr と Cb をスキップさせる。しかし、その時点で色差信号の並びの順番が入れ替わってしまうため、輝度信号と同じ処理を施すことはできない。

【0189】そこで、本実施の形態では、輝度信号についてはキュービック補間を行い、クロマ信号については補間画素近傍のデータから直線補間する手法を用いるようにしている。

【0190】図 12 には、当該手法を用いて 4:2:2 フォーマットのクロマ信号を画素数変換する処理の全体の流れを示す。

れた位相情報とが得られた後は、これら各画素に対応するキュービック係数を求め、畳み込み演算を行う。

【0183】上記キュービック係数は前記式 (1) から計算されるが、実際は  $|x|$  の大きさによって 2 つの場合に分けられ、最終的な出力は以下の式 (17) で与えられる。なお、縮小画素数変換の場合、補間演算により得られた画素データは、以下の式 (10) による値を出力スキップレジスタ 14 に格納された画素スキップ情報に基づいて飛び飛びに出力したものとなる。

【0184】

【0191】この図 12 において、4:2:2 フォーマットのクロマ信号の画素数変換処理を SIMD 制御の基に行うため、ステップ ST 71 にて先ず予めローカルメモリ 15 上に後述するフラグレジスタの領域を確保し、拡大画素数変換又は縮小画素数変換のための補間に必要な情報を計算し、得られた値を当該フラグレジスタに格納する。なお、このステップ ST 71 における拡大画素数変換処理と縮小画素数変換の詳細については後述する。

【0192】次のステップ ST 72 では、ステップ ST 71 にて求めたフラグレジスタの値と、補間画素近傍データと、輝度信号の補間の際に求めた位相情報とを用いて、クロマ信号の直線補間処理を行う。なお、このステップ ST 72 における処理の詳細は後述する。

【0193】図 13 には、例えば拡大画素数変換時の上記図 12 のステップ ST 71 におけるフラグレジスタに格納する情報の計算の手順を示す。

【0194】図 13 において、先ずステップ ST 81 では、上記画素数変換処理を SIMD 制御の基に行うため、全要素プロセッサ 10 において、予めローカルメモリ 15 上にそれぞれ 1 ビット分のフラグレジスタ  $F_{11}$ ,  $F_{12}$ ,  $F_{13}$ ,  $F_{14}$  を確保する。なお、フラグレジスタ  $F_{11}$ ,  $F_{12}$  は、自己の要素プロセッサ 10 に入力されたデータが Cr 信号、Cb 信号の何れに対応するかを示すフラグを格納するためのレジスタであり、例えばあるデータが Cr (R-Y 信号) であれば "1" がフラグレジスタ  $F_{11}$  に設定格納され、Cb (B-Y 信号) であれば "1" がフラグレジスタ  $F_{12}$  に設定格納される。以下の説明では、それぞれ区別するために Cr 入力フラグレジスタ  $F_{11}$ , Cb 入力フラグレジスタ  $F_{12}$  と呼ぶことにする。さらに、これら Cr, Cb がスキップされたときは当該画素に相当する Cr 入力フラグレジスタ  $F_{11}$ , Cb 入力フラグレジスタ  $F_{12}$  に "0" が設定格納される。また、フラグレジスタ  $F_{13}$  に格納されるフラグは、Cr を出力すべき画素に対応し、"1" と "0" が交互に繰り返して格納される。以下の説明では、当該フラグレジスタ  $F_{13}$  を Cr 出力フラグレジスタ  $F_{13}$  と呼ぶことにする。ここでは、当該 Cr 出力フラグ

20

30

40

50

レジスタF<sub>cc</sub>に"1"が格納されたときにはCrを、一方、"0"が格納されたときはCbを出力するものとする。

【0195】次のステップST82では、全要素プロセッサ10において、ローカルメモリ15のフラグレジスタF<sub>1</sub>に、前述の図4の処理にて求めた拡大画素数変換時の画素スキップ情報（入カスキップレジスタ12に格納された画素スキップ情報）をコピーする。なお、画素スキップ情報は前述したように、"1"がスキップすることを、"0"がスキップしないことを表している。以下、この入カスキップレジスタ12から画素スキップ情報がコピーされるローカルメモリ15上のフラグレジスタF<sub>1</sub>を、入カスキップフラグレジスタF<sub>1</sub>と呼ぶことにする。

【0196】次に、ステップST83では、入カスキップフラグレジスタF<sub>1</sub>に格納されている値が"1"か否かの判定を行い、"1"のときにはステップST84に進み、"1"でないとき("0"のとき)にはステップST85に進む。

【0197】上記ステップST83にて入カスキップフラグレジスタF<sub>1</sub>に格納されている値が"1"と判定されたときに進むステップST84では、全要素プロセッサ10において、それぞれ自己の要素プロセッサ10の1つ左隣の要素プロセッサ10のCr入力フラグレジスタF<sub>ic</sub>に格納されている値を、自己の要素プロセッサ10のCr入力フラグレジスタF<sub>ic</sub>に格納する。

【0198】一方、上記ステップST83にて入カスキップフラグレジスタF<sub>1</sub>に格納されている値が"0"と判定されたときに進むステップST85では、全要素プロセッサ10において、それぞれ自己の要素プロセッサ10の1つ左隣の要素プロセッサ10のCr入力フラグレジスタF<sub>ic</sub>に格納されている値の反転した値を、自己の要素プロセッサ10のCr入力フラグレジスタF<sub>ic</sub>に格納する。

【0199】これらステップST84及びステップST85の後には、ステップST86に進む。当該ステップST86では、全要素プロセッサ10において、それぞれ上記ステップST84、ST85にて得られた入カスキップフラグレジスタF<sub>1</sub>の値が"1"となっている要素プロセッサ10ではそれぞれCr入力フラグレジスタF<sub>ic</sub>の値を"0"にする。

【0200】次のステップST87では、全要素プロセッサ10において、それぞれ自己のCr入力フラグレジスタF<sub>ic</sub>に格納している値を反転した値を、同じく自己のCb入力フラグレジスタF<sub>ic</sub>に格納する。

【0201】ステップST88では、全要素プロセッサ10において、それぞれ自己の入カスキップフラグレジスタF<sub>1</sub>の値が"1"となっている要素プロセッサ10ではそれぞれCb入力フラグレジスタF<sub>ic</sub>の値を"0"にする。

【0202】次のステップST89では、全要素プロセッサ10において、それぞれ自己の要素プロセッサ10の1つ左隣の要素プロセッサ10のCr出力フラグレジスタF<sub>oc</sub>に格納している値を反転した値を、自己の要素プロセッサ10内のCr出力フラグレジスタF<sub>oc</sub>に格納する。

【0203】その後、ステップST90では、1ライン分の全画素（少なくとも全要素プロセッサ10の数より多い）について上述の処理が終了したか否かの判定を行い、終了していないと判断したときにはステップST83に戻って上述の処理を繰り返し、終了したと判断したときには次の処理として前記図12のステップST72すなわち次に示す図14のフローチャートの処理に進む。

【0204】なお、最左端の要素プロセッサ10はその左の要素プロセッサが存在しないため各フラグレジスタの値は確定しないことになる。したがって、当該最左端の要素プロセッサ10の場合は、各フラグレジスタに常に"0"を格納することにする。

【0205】次に、拡大画素数変換処理時の上記図12のフローチャートのステップST72における直線補間処理の詳細を、図14のフローチャートを用いて説明する。

【0206】すなわちこの拡大画素数変換処理において、クロマ信号の入力時には、前記輝度信号と同じ画素スキップ情報にてスキップさせて、データをローカルメモリ15に取り込むようにする。このローカルメモリ15上の入力クロマ信号の値と前記フラグレジスタに格納した値とから、直線補間演算を行う。この直線補間では、所望の補間点の左右近傍の2点のデータをサーチし、前記輝度信号の処理時に計算で求めた位相情報に従って線形加算により補間を行う。

【0207】この補間手順を、図14のフローチャートに示す。なお、ここでは、簡略化のため変換比率（拡大比率）を2倍までとして説明している。変換比率が2倍以上の場合は近傍の要素プロセッサとの通信範囲が広がるだけでアルゴリズムは基本的に同じである。また、以下の説明ではCr信号での処理を中心にして述べているが、Cb信号についても同様に行う。この場合、上記図14のフローチャートにおいて、後述する各ステップにおけるCrライトレジスタd<sub>cc</sub>をCbライトレジスタd<sub>cc</sub>に置き換え、以下同様に、Crレフトレジスタd<sub>cc</sub>をCbレフトレジスタd<sub>cc</sub>に、Cr入力フラグレジスタF<sub>ic</sub>をCb入力フラグレジスタF<sub>ic</sub>に置き換えて説明すればよい。

【0208】図14において、ステップST111ではCr信号についての各補間画素の最左隣画素値と最右隣画素値を格納する領域としてローカルメモリ15上にCrライトレジスタd<sub>cc</sub>、Crレフトレジスタd<sub>cc</sub>を確保する。なお、Cb信号については各補間画素の最左隣

画素値と最右隣画素値を格納する領域としてローカルメモリ15上にCbライトレジスタd<sub>cr</sub>, Cbレフトレジスタd<sub>cl</sub>を確保する。

【0209】ステップST112では、4:2:2フォーマットのクロマ信号をローカルメモリ15上に確保したクロマ入力レジスタd<sub>ci</sub>に格納する。

【0210】ステップST113では、上記全要素プロセッサ10において、それぞれ4つ左隣の要素プロセッサ10のクロマ入力レジスタd<sub>ci</sub>に格納している値を、自己の要素プロセッサのCrレフトレジスタd<sub>cl</sub>に格納する。

【0211】次のステップST114では、全要素プロセッサ10において、それぞれ3つ左隣の要素プロセッサ10の前記Cr入力フラグレジスタF<sub>icr</sub>の値が“1”か否かを判定する。当該ステップST114において、3つ左隣の要素プロセッサ10のCr入力フラグレジスタF<sub>icr</sub>の値が“1”であるときにはステップST115に進み、“1”でないとき(“0”のとき)はそのままステップST116に進む。

【0212】上記ステップST114にて3つ左隣の要素プロセッサ10の前記Cr入力フラグレジスタF<sub>icr</sub>の値が“1”であると判定されたときのステップST115では、上記3つ左隣の要素プロセッサ10のクロマ入力レジスタd<sub>ci</sub>に格納されている値を、自己の要素プロセッサ10のCrレフトレジスタd<sub>cl</sub>に格納する。

【0213】上記ステップST114にて3つ左隣の要素プロセッサ10の前記Cr入力フラグレジスタF<sub>icr</sub>の値が“0”であると判定されたとき、及び、上記ステップST115の処理後に進む、ステップST116では、全要素プロセッサ10において、それぞれ2つ左隣の要素プロセッサ10の前記Cr入力フラグレジスタF<sub>icr</sub>の値が“1”か否かを判定する。当該ステップST116において、2つ左隣の要素プロセッサ10のCr入力フラグレジスタF<sub>icr</sub>の値が“1”であるときにはステップST117に進み、“1”でないとき(“0”のとき)はそのままステップST118に進む。

【0214】上記ステップST116にて2つ左隣の要素プロセッサ10の前記Cr入力フラグレジスタF<sub>icr</sub>の値が“1”であると判定されたときのステップST117では、上記2つ左隣の要素プロセッサ10のクロマ入力レジスタd<sub>ci</sub>に格納されている値を、自己の要素プロセッサ10のCrレフトレジスタd<sub>cl</sub>に格納する。

【0215】上記ステップST116にて2つ左隣の要素プロセッサ10の前記Cr入力フラグレジスタF<sub>icr</sub>の値が“0”であると判定されたとき、及び、上記ステップST117の処理後に進む、ステップST118では、全要素プロセッサ10において、それぞれ1つ左隣の要素プロセッサ10の前記Cr入力フラグレジスタF<sub>icr</sub>の値が“1”か否かを判定する。当該ステップST118において、1つ左隣の要素プロセッサ10のCr

入力フラグレジスタF<sub>icr</sub>の値が“1”であるときにはステップST119に進み、“1”でないとき(“0”のとき)はそのまま図15に示す次の処理に進む。

【0216】上記ステップST118にて1つ左隣の要素プロセッサ10の前記Cr入力フラグレジスタF<sub>icr</sub>の値が“1”であると判定されたときのステップST119では、上記1つ左隣の要素プロセッサ10のクロマ入力レジスタd<sub>ci</sub>に格納されている値を、自己の要素プロセッサ10のCrレフトレジスタd<sub>cl</sub>に格納する。

【0217】上記ステップST118にて1つ左隣の要素プロセッサ10の前記Cr入力フラグレジスタF<sub>icr</sub>の値が“0”であると判定されたとき、及び、上記ステップST119の処理後に進む、図15のフローチャートのステップST121では、全要素プロセッサ10において、それぞれ自己の要素プロセッサ10の前記Cr入力フラグレジスタF<sub>icr</sub>の値が“1”か否かを判定する。当該ステップST121において、自己の要素プロセッサ10のCr入力フラグレジスタF<sub>icr</sub>の値が“1”であるときにはステップST122に進み、“1”でないとき(“0”のとき)はそのままステップST123の処理に進む。

【0218】上記ステップST121にて自己の要素プロセッサ10の前記Cr入力フラグレジスタF<sub>icr</sub>の値が“1”であると判定されたときのステップST122では、上記自己の要素プロセッサ10のクロマ入力レジスタd<sub>ci</sub>に格納されている値を、自己の要素プロセッサ10のCrレフトレジスタd<sub>cl</sub>に格納する。

【0219】上記ステップST122にて自己の要素プロセッサ10の前記Cr入力フラグレジスタF<sub>icr</sub>の値が“0”であると判定されたとき、及び、上記ステップST122の処理後に進む、ステップST123では、上記全要素プロセッサ10において、それぞれ4つ右隣の要素プロセッサ10のクロマ入力レジスタd<sub>ci</sub>に格納している値を、自己の要素プロセッサのCrライトレジスタd<sub>cr</sub>に格納する。

【0220】次のステップST124では、全要素プロセッサ10において、それぞれ3つ右隣の要素プロセッサ10の前記Cr入力フラグレジスタF<sub>icr</sub>の値が“1”か否かを判定する。当該ステップST124において、3つ右隣の要素プロセッサ10のCr入力フラグレジスタF<sub>icr</sub>の値が“1”であるときにはステップST125に進み、“1”でないとき(“0”のとき)はそのままステップST126に進む。

【0221】上記ステップST124にて3つ右隣の要素プロセッサ10の前記Cr入力フラグレジスタF<sub>icr</sub>の値が“1”であると判定されたときのステップST125では、上記3つ右隣の要素プロセッサ10のクロマ入力レジスタd<sub>ci</sub>に格納されている値を、自己の要素プロセッサ10のCrライトレジスタd<sub>cr</sub>に格納する。

【0222】上記ステップST124にて3つ右隣の要

要素プロセッサ 10 の前記 Cr 入力フラグレジスタ F<sub>icr</sub> の値が "0" であると判定されたとき、及び、上記ステップ ST 125 の処理後に進む、ステップ ST 126 では、全要素プロセッサ 10 において、それぞれ 2 つ右隣の要素プロセッサ 10 の前記 Cr 入力フラグレジスタ F<sub>icr</sub> の値が "1" か否かを判定する。当該ステップ ST 126 において、2 つ右隣の要素プロセッサ 10 の Cr 入力フラグレジスタ F<sub>icr</sub> の値が "1" であるときにはステップ ST 127 に進み、"1" でないとき ("0" のとき) はそのままステップ ST 128 に進む。

【0223】上記ステップ ST 126 にて 2 つ右隣の要素プロセッサ 10 の前記 Cr 入力フラグレジスタ F<sub>icr</sub> の値が "1" であると判定されたときのステップ ST 127 では、上記 2 つ右隣の要素プロセッサ 10 のクロマ入力レジスタ d<sub>cr</sub> に格納されている値を、自己の要素プロセッサ 10 の Cr ライトレジスタ d<sub>cr</sub> に格納する。

【0224】上記ステップ ST 126 にて 2 つ右隣の要素プロセッサ 10 の前記 Cr 入力フラグレジスタ F<sub>icr</sub> の値が "0" であると判定されたとき、及び、上記ステップ ST 127 の処理後に進む、ステップ ST 128 では、全要素プロセッサ 10 において、それぞれ 1 つ右隣の要素プロセッサ 10 の前記 Cr 入力フラグレジスタ F<sub>icr</sub> の値が "1" か否かを判定する。当該ステップ ST 128 において、1 つ右隣の要素プロセッサ 10 の Cr 入力フラグレジスタ F<sub>icr</sub> の値が "1" であるときにはステップ ST 129 に進み、"1" でないとき ("0" のとき) はそのまま図 16 に示す次の処理に進む。

【0225】上記ステップ ST 128 にて 1 つ右隣の要素プロセッサ 10 の前記 Cr 入力フラグレジスタ F<sub>icr</sub> の値が "1" であると判定されたときのステップ ST 129 では、上記 1 つ右隣の要素プロセッサ 10 のクロマ入力レジスタ d<sub>cr</sub> に格納されている値を、自己の要素プロセッサ 10 の Cr ライトレジスタ d<sub>cr</sub> に格納する。

【0226】上記ステップ ST 128 にて 1 つ右隣の要素プロセッサ 10 の前記 Cr 入力フラグレジスタ F<sub>icr</sub> の値が "0" であると判定されたとき、及び、上記ステップ ST 129 の処理後に進む、図 16 のフローチャートのステップ ST 131 では、全要素プロセッサ 10 において、それぞれ自己の要素プロセッサ 10 の前記 Cr 入力フラグレジスタ F<sub>icr</sub> の値が "1" か否かを判定する。当該ステップ ST 131 において、自己の要素プロセッサ 10 の Cr 入力フラグレジスタ F<sub>icr</sub> の値が "1" であるときにはステップ ST 132 に進み、"1" でないとき ("0" のとき) はそのままステップ ST 133 の処理に進む。

【0227】上記ステップ ST 131 にて自己の要素プロセッサ 10 の前記 Cr 入力フラグレジスタ F<sub>icr</sub> の値が "1" であると判定されたときのステップ ST 132 では、上記自己の要素プロセッサ 10 のクロマ入力レジスタ d<sub>cr</sub> に格納されている値を、自己の要素プロセッサ

10 の Cr ライトレジスタ d<sub>cr</sub> に格納する。

【0228】上記ステップ ST 132 にて自己の要素プロセッサ 10 の前記 Cr 入力フラグレジスタ F<sub>icr</sub> の値が "0" であると判定されたとき、及び、上記ステップ ST 132 の処理後に進む、ステップ ST 133 では、全要素プロセッサ 10 において、前記 Cr 信号についての補間データの最左隣画素値を格納する Cr レフトレジスタ d<sub>cr</sub> と最右隣画素値を格納する Cr ライトレジスタ d<sub>cr</sub> を、前記輝度信号の計算で求めた位相情報に従って、線形加算による補間演算を行う。

【0229】次のステップ ST 134 は Cb 信号の場合の処理を行う、すなわち、当該 Cb 信号についても、前記図 14 のステップ ST 112 ~ 図 16 のステップ ST 143 における前記 Cr ライトレジスタ d<sub>cr</sub> を Cb ライトレジスタ d<sub>cr</sub> に置き換え、Cr レフトレジスタ d<sub>cr</sub> を Cb レフトレジスタ d<sub>cr</sub> に、Cr 入力フラグレジスタ F<sub>icr</sub> を Cb 入力フラグレジスタ F<sub>icb</sub> に置き換えて、前述同様の処理を行い、Cb 信号について線形加算補間演算による補間データを求める。

【0230】次のステップ ST 135 では、全要素プロセッサ 10 において、前記 Cr 出力フラグレジスタ F<sub>ocr</sub> の値が "1" であるか否かの判定を行い、当該 Cr 出力フラグレジスタ F<sub>ocr</sub> の値が "1" のときにはステップ ST 137 に進み、"1" でないとき ("0" のとき) はステップ ST 136 に進む。

【0231】上記 Cr 出力フラグレジスタ F<sub>ocr</sub> の値が "1" である場合に進むステップ ST 137 では上記ステップ ST 133 にて求めた Cr 信号についての補間データを出力し、逆に Cr 出力フラグレジスタ F<sub>ocr</sub> の値が "0" である場合に進むステップ ST 136 では上記ステップ ST 134 にて求めた Cb 信号についての補間データを出力し、これら Cr 信号についての補間データと Cb 信号についての補間データを画素毎に選択して前記ローカルメモリ 15 上に設けた 4:2:2 フォーマットの Cr、Cb 信号用出力レジスタに格納する。

【0232】次に、前述した拡大画素数変換時には、画素データ入力時に画素データが飛び飛びに入力レジスタ 11 に格納され、出力時には出力レジスタ 13 からそのままストレートに出力されたのに対して、縮小画素数変換では画素データ入力時にはデータが入力レジスタ 11 にそのままストレートに入力され、出力時に出力レジスタ 13 から飛び飛びに出力される点異なる。この縮小画素数変換における補間処理については前記拡大画素数変換のときと同じであるので説明は省略し、ここでは補間演算の前処理部分について述べる。

【0233】この縮小画素数変換の場合も、全要素プロセッサ 10 において、先ず予めローカルメモリ 15 上に前述同様の Cr 入力フラグレジスタ F<sub>icr</sub> と Cb 入力フラグレジスタ F<sub>icb</sub> と Cr 出力フラグレジスタ F<sub>ocr</sub> とを確保する。また当該縮小画素数変換時には、出力スキッ

10

20

30

40

50

レジスタ 14 に格納された画素スキップ情報を格納するための出力スキップフラグレジスタ  $F_{os}$  をローカルメモリ 15 上に確保する。これら各フラグレジスタはそれぞれ 1 ビットレジスタである。

【0234】なお、上記入力  $C_r$  フラグレジスタ  $F_{ic}$  と入力  $C_b$  フラグレジスタ  $F_{ib}$  は、前述同様に、自己の要素プロセッサ 10 に入力されたデータが  $C_r$  信号、 $C_b$  信号の何れに対応するかを示すフラグを格納するためのレジスタであり、例えばあるデータが  $C_r$  であれば “1” がフラグレジスタ  $F_{ic}$  に設定格納され、 $C_b$  であれば “1” がフラグレジスタ  $F_{ib}$  に設定格納される。縮小画素数変換の場合には、これら  $C_r$  入力フラグレジスタ  $F_{ic}$  と  $C_b$  入力フラグレジスタ  $F_{ib}$  には、画素毎に “1” と “0” が交互に繰り返された値が格納され、 $C_r$  入力フラグレジスタ  $F_{ic}$  と  $C_b$  入力フラグレジスタ  $F_{ib}$  の対応する画素について言えば反転したものととなっている。また、 $C_r$  出力フラグレジスタ  $F_{oc}$  には、前述同様に  $C_r$  を出力すべき画素に対応して “1” と “0” が交互に繰り返す値が格納され、当該  $C_r$  出力フラグレジスタ  $F_{oc}$  に “1” が格納されたときには  $C_r$  を、一方、“0” が格納されたときは  $C_b$  を出力するものとする。

【0235】当該縮小画素数変換の処理を SIMD 制御で行うと、前記図 12 のフローチャートのステップ ST 71 におけるフラグレジスタに格納する情報の計算の手順は、以下の図 17 のフローチャートに示すような流れとなる。

【0236】この図 17 において、先ずステップ ST 101 では、全要素プロセッサ 10 において、予めローカルメモリ 15 上にそれぞれ 1 ビット分の出力スキップフラグレジスタ  $F_{os}$  と  $C_r$  入力フラグレジスタ  $F_{ic}$  と  $C_b$  入力フラグレジスタ  $F_{ib}$  と  $C_r$  出力フラグレジスタ  $F_{oc}$  とを確保する。

【0237】次のステップ ST 102 では、全要素プロセッサ 10 においてそれぞれローカルメモリ 15 の出力スキップフラグレジスタ  $F_{os}$  に、前述の図 9 の処理にて求めた縮小画素数変換時の画素スキップ情報（出力スキップレジスタ 14 に格納された画素スキップ情報）をコピーする。なお、この画素スキップ情報は前述したように、“1” がスキップすることを、“0” がスキップしないことを表している。

【0238】次に、ステップ ST 103 では、全要素プロセッサ 10 においてそれぞれ出力スキップフラグレジスタ  $F_{os}$  に格納されている値が “1” か否かの判定を行い、“1” のときにはステップ ST 104 に進み、“1” でないとき (“0” のとき) にはステップ ST 105 に進む。

【0239】上記ステップ ST 103 にて出力スキップフラグレジスタ  $F_{os}$  に格納されている値が “1” と判定されたときに進むステップ ST 104 では、全要素プロ

セッサ 10 において、それぞれ自己の要素プロセッサ 10 の 1 つ左隣の要素プロセッサ 10 の  $C_r$  出力フラグレジスタ  $F_{oc}$  に格納されている値を、自己の要素プロセッサ 10 の  $C_r$  出力フラグレジスタ  $F_{oc}$  に格納する。

【0240】一方、上記ステップ ST 103 にて出力スキップフラグレジスタ  $F_{os}$  に格納されている値が “0” と判定されたときに進むステップ ST 105 では、全要素プロセッサ 10 において、それぞれ自己の要素プロセッサ 10 の 1 つ左隣の要素プロセッサ 10 の  $C_r$  出力フラグレジスタ  $F_{oc}$  に格納されている値の反転した値を、自己の要素プロセッサ 10 の  $C_r$  出力フラグレジスタ  $F_{oc}$  に格納する。

【0241】これらステップ ST 104 及びステップ ST 105 の後は、ステップ ST 106 に進む。当該ステップ ST 106 では、1 ライン分の全画素（少なくとも全要素プロセッサ 10 の数より多い）について上述の処理が終了したか否かの判定を行い、終了していないと判断したときにはステップ ST 103 に戻って上述の処理を繰り返し、終了したと判断したときには次の処理、すなわち前述同様の図 14 から図 16 の各フローチャートの処理に進む。ただし、このときの図 14 から図 16 のフローチャートでは、上記図 17 の処理にて求めた  $C_r$  入力フラグレジスタ  $F_{ic}$  と  $C_b$  入力フラグレジスタ  $F_{ib}$  と  $C_r$  出力フラグレジスタ  $F_{oc}$  の値を用いて前述同様の処理を行い、当該縮小画素数変換処理ときの 4 : 2 : 2 フォーマットのクロマ信号出力を得るようにする。

【0242】なお、この縮小画素数変換処理の場合も、最左端の要素プロセッサ 10 はその左の要素プロセッサが存在しないため各フラグレジスタの値は確定しないことになる。したがって、当該最左端の要素プロセッサ 10 の場合は、各フラグレジスタに常に “0” を格納することにする。

【0243】上述したような手法による 4 : 2 : 2 フォーマットにおける拡大画素数変換と縮小画素数変換の他に、当該 4 : 2 : 2 フォーマットのクロマ信号を一旦 4 : 4 : 4 フォーマットに変換して拡大画素数変換や縮小画素数変換を行うことも可能である。

【0244】この 4 : 2 : 2 フォーマットから 4 : 4 : 4 フォーマットへのフォーマット変換の流れは、例えば図 18 に示すようなフローチャートとなる。

【0245】この図 18 において、ステップ ST 73 では 4 : 2 : 2 フォーマットのクロマ信号を 4 : 4 : 4 フォーマットに変換し、次のステップ ST 74 では当該 4 : 4 : 4 フォーマットのクロマ信号を前述したようにして拡大画素数変換や縮小画素数変換する。

【0246】ここで、上記ステップ ST 73 における 4 : 2 : 2 フォーマットから 4 : 4 : 4 フォーマットへのフォーマット変換例としては図 19 に示すような手法が考えられる。すなわち、この図 19 では、 $C_r$  判別フ



ラグとCb判別フラグを用い、Cb判別フラグと4:2:2クロマ信号との論理積演算後のCr信号と、Cr判別フラグと4:2:2クロマ信号との論理積演算後のCb信号とを求め、これら論理積演算後のCr信号とCb信号から4:4:4フォーマットのCr信号及びCb信号を演算により求めるようにしている。

【0247】上記SIMD制御のリニアアレイ型多並列プロセッサ1において、上記図19に示すようなフォーマット変換を実現する場合、図20のフローチャートに示すような手順となる。

【0248】図20において、ステップST141では、上記4:2:2フォーマットのCr信号を判別する上記Cr判別フラグと、Cb信号を判別するCb判別フラグとを、ローカルメモリ15上に用意する。これらCr判別フラグ、Cb判別フラグは、当該リニアアレイ型多並列プロセッサ1内部で生成してローカルメモリ15上に用意してもよいし、また、前述したCr、Cb用のフラグの生成にて説明したと同様のナンバリング手法にて生成することを可能である。

【0249】次のステップST142では、これらCr判別フラグとCb判別フラグそれぞれに対して図19の4:2:2フォーマットのクロマ信号との論理積(AND)演算を行う。

【0250】ステップST143では当該論理積演算により得られたCr信号と0が交互に繰り返す信号と、Cb信号と0が交互に繰り返す信号を、ローカルメモリ15上に保存する。

【0251】次のステップST144では、上記ステップST142及びステップST143にて得られたそれぞれの信号に、1/2, 1, 1/2の3タップのFIRフィルタリングを施す。これは、単に平均補間であるが、輝度信号に対して4:2:2フォーマットのクロマ信号では元々周波数帯域が狭いので問題は起きない。

【0252】このようにして得られた4:4:4フォーマットの信号をステップST145にて出力する。

【0253】次に、4:1:1フォーマットのクロマ信号の画素数変換について説明する。

【0254】この4:1:1フォーマットのクロマ信号の画素数変換時には、図21のフローチャートに示すように、前記リニアアレイ型多並列プロセッサ1において、ステップST75にて当該4:1:1フォーマットのクロマ信号を一旦4:2:2フォーマットのクロマ信号に変換し、次のステップST76にて上記ステップST75で得られた4:2:2フォーマットのクロマ信号に対して、前述同様の画素数変換処理を施すようにする。

【0255】上記図21のステップST75における4:1:1フォーマットから4:2:2フォーマットへの変換処理の流れは、図22～図24のフローチャートに示すようになる。また、これら図22～図24に示す

処理におけるローカルメモリ15の内容は、図25及び図26に示すようになる。なお、図25と図26は、図25の後に図26が続いた1枚の図として表すべきものであるが、紙面の関係で2つの図として表している。以下、これら図22～図24の処理を図25及び図26を参照して説明する。

【0256】先ず、図22に示すステップST151では、全要素プロセッサ10においてそれぞれローカルメモリ15上に4:1:1フォーマットのクロマ信号の最上位ビット(MSB)位置を表すための1ビットの判別フラグを格納する判別フラグレジスタd<sub>11</sub>を確保する。この判別フラグは、図25に判別フラグ信号として示す4サイクルの周期信号であり、例えば当該リニアアレイ型多並列プロセッサ1の外部で生成して入力してもよいし、前記ナンバリング処理にて当該プロセッサ1内部で生成してもよい。当該リニアアレイ型多並列プロセッサ1内部のナンバリング処理にて当該判別フラグを生成した場合、上記判別フラグレジスタd<sub>11</sub>を当該プロセッサ1内部にて各要素プロセッサ10間で通信して、左右に移動して4:1:1フォーマットのクロマ信号のMSB位置と一致させる。また、当該ステップST151では、ローカルメモリ15上に作業アドレス空間として3ビットのレジスタ(以下、作業用レジスタd<sub>12</sub>と呼ぶ)をも確保する。

【0257】次に、ステップST152では、全要素プロセッサ10において、それぞれ1つ左隣の要素プロセッサ10のローカルメモリ15上の上記作業用レジスタd<sub>12</sub>の値に1を加算した値を、自己の要素プロセッサの作業用レジスタd<sub>12</sub>に格納する。

【0258】次のステップST153では、1ライン分の全画素(少なくとも全要素プロセッサ10の数より多い)について上述の処理が終了したか否かの判定を行い、終了していないと判断したときにはステップST152に戻って上述の処理を繰り返し、終了したと判断したときにはステップST154に進む。

【0259】次のステップST154では、上記判別フラグの値として、上記3ビット作業用レジスタd<sub>12</sub>の下位2ビットの各ビットの否定論理和演算を行い、得られた値を上記判定フラグとして、ローカルメモリ15上に確保した判定フラグレジスタd<sub>13</sub>に格納する。なお、このステップST154における判定フラグの決定処理をより具体的に説明すると、上記3ビットの作業用レジスタd<sub>12</sub>の下位2ビットの値が"00"であるか否かの判断を行い、当該下位2ビットの値が"00"であるときには上記判別フラグを"1"とし、当該下位2ビットの値が"00"でないときには判別フラグを"0"とする。ただし、最左端の要素プロセッサ10はその左の要素プロセッサが存在しないため判別フラグの値は確定しないことになる。したがって、当該最左端の要素プロセッサ10の場合は、判別フラグとして常に"



0" が与えられるものとする。

【0260】次のステップST155では、全要素プロセッサ10においてそれぞれ上記ローカルメモリ15上に8ビットの4:2:2フォーマットのクロマ信用のクロマレジスタd...と、それぞれが8ビットの作業用レジスタWO<sub>i</sub>、WO<sub>j</sub>を確保する。

【0261】次のステップST156では、全要素プロセッサ10において上記ローカルメモリ15上にそれぞれ4:1:1フォーマットの入力Cr信号と入力Cb信号それぞれ2ビットずつを格納する。

【0262】すなわちステップST157では、全要素プロセッサ10において、それぞれ3つ右隣の要素プロセッサ10のローカルメモリ15に格納された上記4:1:1フォーマットの入力Cr信号2ビットを、自己の要素プロセッサ10の上記8ビット作業用レジスタWO<sub>i</sub>の第0ビット(図25ではWO<sub>i</sub>として示す)と第1ビット(図25ではWO<sub>i</sub>+1として示す)に格納する。

【0263】次のステップST158では、同様に、全要素プロセッサ10において、それぞれ2つ右隣の要素プロセッサ10のローカルメモリ15に格納された上記4:1:1フォーマットの入力Cr信号2ビットを、自己の要素プロセッサ10の上記8ビット作業用レジスタWO<sub>i</sub>の第2ビット(図25ではWO<sub>i</sub>+2として示す)と第3ビット(図25ではWO<sub>i</sub>+3として示す)に格納する。このステップST158の処理御は、図23のフローチャートのステップST161に進む。

【0264】当該図23のステップST161では、同様に、全要素プロセッサ10において、それぞれ1つ右隣の要素プロセッサ10のローカルメモリ15に格納された上記4:1:1フォーマットの入力Cr信号2ビットを、自己の要素プロセッサ10の上記8ビット作業用レジスタWO<sub>i</sub>の第4ビット(図25ではWO<sub>i</sub>+4として示す)と第5ビット(図25ではWO<sub>i</sub>+5として示す)に格納する。

【0265】次の、ステップST162では、全要素プロセッサ10において、自己の要素プロセッサのローカルメモリ15に格納された上記4:1:1フォーマットの入力Cr信号2ビットを、自己の要素プロセッサ10の上記8ビット作業用レジスタWO<sub>i</sub>の第6ビット(図25ではWO<sub>i</sub>+6として示す)と第7ビット(図25ではWO<sub>i</sub>+7として示す)に格納する。

【0266】ここまでの処理により、ローカルメモリ15内の作業用レジスタWO<sub>i</sub>には入力Cr信号について図25の(a)に示すような値が格納されることになる。

【0267】以下、作業用レジスタWO<sub>i</sub>についても同様に、ステップST163では、全要素プロセッサ10において、それぞれ3つ右隣の要素プロセッサ10のローカルメモリ15に格納された上記4:1:1フォーマットの入力Cb信号2ビットを、自己の要素プロセッサ

10の上記8ビット作業用レジスタWO<sub>i</sub>の第0ビットと第1ビットに格納する。

【0268】ステップST164では、全要素プロセッサ10において、それぞれ2つ右隣の要素プロセッサ10のローカルメモリ15に格納された上記4:1:1フォーマットの入力Cb信号2ビットを、自己の要素プロセッサ10の上記8ビット作業用レジスタWO<sub>i</sub>の第2ビットと第3ビットに格納する。

【0269】ステップST165では、全要素プロセッサ10において、それぞれ2つ右隣の要素プロセッサ10のローカルメモリ15に格納された上記4:1:1フォーマットの入力Cb信号2ビットを、自己の要素プロセッサ10の上記8ビット作業用レジスタWO<sub>i</sub>の第4ビットと第5ビットに格納する。

【0270】ステップST166では、全要素プロセッサ10において、自己の要素プロセッサ10のローカルメモリ15に格納された上記4:1:1フォーマットの入力Cb信号2ビットを、自己の要素プロセッサ10の上記8ビット作業用レジスタWO<sub>i</sub>の第6ビットと第7ビットに格納する。

【0271】次に、ステップST167では、全要素プロセッサ10において、上述のようにして格納された作業用レジスタWO<sub>i</sub>の8ビットのそれぞれのビットと、前記判別フラグレジスタd...に格納された判別フラグビットとの間で、論理積(AND)演算を行い、得られた値をそれぞれ作業用レジスタWO<sub>i</sub>に戻す。

【0272】この処理により、ローカルメモリ15内の例えば作業用レジスタWO<sub>i</sub>には図25の(b)に示すような値が格納されることになる。

【0273】次の、ステップST169では、同様にして、全要素プロセッサ10において、上述のようにして格納された作業用レジスタWO<sub>i</sub>の8ビットのそれぞれのビットと、前記判別フラグレジスタd...に格納された判別フラグビットとの間で、論理積(AND)演算を行い、得られた値をそれぞれ作業用レジスタWO<sub>i</sub>に戻す。

【0274】その後、図24のフローチャートのステップST171では、全要素プロセッサ10において、上記作業用レジスタWO<sub>i</sub>の8ビットに対して、1/2, 0, 1, 0, 1/2のタップのFIRフィルタリング処理を施す。

【0275】ここまでの処理により、ローカルメモリ15内の例えば作業用レジスタWO<sub>i</sub>には図25の(c)に示すような値が格納されることになる。

【0276】同様に、ステップST172では、全要素プロセッサ10において、上記作業用レジスタWO<sub>i</sub>の8ビットに対して、1/2, 0, 1, 0, 1/2のタップのFIRフィルタリング処理を施す。

【0277】次のステップST173では、上記作業用レジスタWO<sub>i</sub>の値を、前記4:2:2フォーマットの

10

20

30

40

50

クロマレジスタ  $d_{c...}$  に格納する。

【0278】最後に、全要素プロセッサ10において、それぞれ1つ左隣の要素プロセッサの作業用レジスタ  $WO_i$  の値と自己の要素プロセッサ10の上記クロマレジスタ  $d_{c...}$  の値との論理和 (OR) 演算を行い、得られた値を自己の要素プロセッサの上記クロマレジスタ  $d_{c...}$  に戻す。

【0279】ここまでの処理により、ローカルメモリ15内の4:2:2フォーマットのクロマレジスタ  $d_{c...}$  には図26に示すような値が格納されることになる。なお、このクロマレジスタ  $d_{c...}$  においても、8ビットを前記図25の作業用レジスタ  $WO_i$  の場合と同様に、第0ビットを  $d_{c...}$  と表し、第1ビットを  $d_{c...} + 1$  と表し、第2ビットを  $d_{c...} + 2$  と表し、以下同様にして第7ビットを  $d_{c...} + 7$  と表している。

【0280】その後は、図21のステップST76の処理に進む。

【0281】次に、上記図24のステップST171及びステップST172におけるFIRフィルタの構成は、図27に示すようなものを挙げることができる。

【0282】この図27において、入力端子50にはクロマのデータが供給され、直列接続された1サンプル遅延器51~54に順次送られる。これら遅延器51~54は、それぞれ供給されたデータを1サンプル分づつ遅延するものである。上記入力端子50の入力データとこれら遅延器51~54の各出力データは、それぞれ対応する乗算器55~59に送られる。

【0283】これら乗算器55~59には、それぞれ1/2, 0, 1, 0, 1/2の乗算係数(フィルタ係数)が設定されている。したがって、各乗算器55~59では、上記乗算係数と、上記入力端子60への入力データと各遅延器51~54の各出力データとをかけ算する。この乗算器55~59の各乗算結果は、加算器60により加算され出力端子61からFIRフィルタ出力として取り出される。

【0284】この図27に示すFIRフィルタは、各要素プロセッサ10間において近傍通信を行うことにより容易に実現できるものである。

【0285】上述のようにして、4:1:1フォーマットのクロマ信号を4:2:2フォーマットのクロマ信号に変換すれば、その後は前述した4:2:2フォーマットにおける画素数変換のアルゴリズムを使用するか、或いはこの4:2:2フォーマットをさらに4:4:4フォーマットに変換して前述した4:4:4フォーマットにおける画素数変換アルゴリズムを使用すれば、当該4:1:1フォーマットの画素数変換が可能となる。

【0286】なお、上述した説明では、拡大画素数変換、縮小画素数変換処理の実施の形態について述べてきたが、これらの技術は、画素データの入力レートと出力レートをそれぞれ独立に制御することにより、サンプリ

ング周波数変換処理にもそのまま適応可能である。

【0287】次に走査線数変換処理について説明する。

【0288】本発明実施の形態の走査線数変換処理では、本発明の特徴の一つであるリアルタイムな走査線数変換も可能である。

【0289】映像信号の水平方向のサンプル点を垂直方向に眺めて、各画素をラインに置き換えれば、走査線数変換となり、いままでの画素数変換と同じ考え方をち適用することができる。この場合は入出力のスキップ機能は、外部のフィールドFIFOメモリ等で調整すればよく、この制御用信号はライン属性情報に基づいてライン属性情報の計算と同時に求めることが可能である。

【0290】また、演算処理自体は輝度信号とクロマ信号とを区別する必要はなく、同じ処理で行うことができる。

【0291】画素数変換処理では、画素というものを単位で行っていた訳だが、これをライン単位と置き換えれば走査線数変換処理も同じように扱うことができる。すなわち、入力ライン近傍の4ラインデータとライン位相情報とのキュービク演算から補間すべきラインを計算する。ただし、画素数変換処理においては、前記図3のステップST2に示す画素スキップ情報と画素位相情報をブランキング区間或いは電源投入時等に行っていたが、走査線数変換処理では入力スキップライン情報、出力スキップライン情報、ライン位相情報は毎ライン計算する点と、前記ライン位相情報については拡大用と縮小用と2種類の位相情報のためのレジスタを持つ点が大きく異なる。このように、ライン位相情報を2種類持つことで縮小から拡大まで任意の比率変換を可能としている。また、この2種類の位相情報から求められる入力スキップライン制御信号と上記スキップライン制御信号で外部のフィールドメモリと当該DSP自体の処理プログラムを制御することにより、補間演算に必要なDSP内部のローカルメモリ量を削減している。

【0292】図28に当該DSPを用いて画素数変換と走査線数変換処理を行う場合のブロック構成の一例を示す。この場合、当該DSPのリニアアレイ型多並列プロセッサ72において、図中の点線にて囲んだ構成(画素数変換部71)にて画素数変換がなされ、それ以外の構成にて走査線数変換処理が行われる。当該DSPはそもそもハードウェアではなく、ソフトウェアだけで構成されるため、実際の実現方法は異なるが、内部の各信号処理の一つ一つをブロック分けすると図28のように表すことができ、画素数変換部71とそれ以外の走査線数変換部とに分けることが可能であり、画素数変換については既に説明してあるので、以下では走査線数変換処理についてその構成及び動作を説明する。

【0293】具体的な走査線数変換を実現する処方を説明する。ただし、この場合もリニアアレイ型多並列プロセッサは画素数変換で使ったものと同じであるため、こ

ここではリニアアレイ型多並列プロセッサの構成についての説明は簡略化する。

【0294】この図28において、リニアアレイ型多並列プロセッサ72の画素数変換部71は、前述したように、出力スキップ画素を計算する出力スキップ画素計算部52と入力スキップ画素を計算する入力スキップ画素計算部53、及び輝度、クロマ用のフィルタ部54、画素位相計算部55とからなる。

【0295】また、リニアアレイ型多並列プロセッサ72の走査線数変換部は、縮小用ライン位相計算部58及び拡大用ライン位相計算部62、出力スキップライン計算部60及び入力スキップライン計算部63、ライン位相用レジスタ59、出力スキップライン用レジスタ61、信号用ディレイライン56、フィルタ部57とからなる。上記縮小用ライン位相計算部58及び拡大用ライン位相計算部62、出力スキップライン計算部60及び入力スキップライン計算部63、ライン位相用レジスタ59、出力スキップライン用レジスタ61は、前述した画素数変換処理のための構成と対応したものであるが、これらではライン単位での処理を行う。

【0296】データ入力端子50からは、輝度とクロマの信号が入力され、これらの信号IRは外部のフィールドメモリ51を介して、リニアアレイ型多並列プロセッサ72に供給される。当該フィールドメモリ51から供給された信号IRは、走査線数変換部の信号用ディレイライン56に供給される。この信号用ディレイライン56は、上記信号IRを4H分（4水平周期分）遅延させて各ライン毎に出力するものである。これらライン毎の信号は上記フィルタ部54と同様のフィルタ部57に送られる。このフィルタ部57の出力は画素数変換部71のフィルタ部54を介し、出力信号ORとして外部フィールドメモリ64に送られる。このフィールドメモリ64からの輝度及びクロマの信号が画素数或いはライン数変換された出力としてデータ出力端子65から出力される。

【0297】また、上記入力スキップライン計算部63からは、フィールドメモリ51への後述する制御信号として入力スキップライン制御信号F<sub>sk</sub>が出力され、出力スキップライン用レジスタ61からは、フィールドメモリ64への後述する制御信号として出力スキップライン制御信号F<sub>sk</sub>が出力される。ライン位相用レジスタ59からはフィルタ部57の制御信号が出力される。

【0298】図29、図30、図31を用いて、上記走査線数変換部における処理の全体の流れを説明する。

【0299】図29において、まず、ステップST200で走査線数変換の変換比率N:Mが設定される。ただし、N及びMは、正の正数であり、M≧Nで拡大走査線数変換、M<Nで縮小走査線数変換となる。変換比率1:1の等倍変換は拡大変換に含めることとする。

【0300】次のステップST201は、画像の1フレ

ームの最初のラインに実行されるが、ここで入力スキップラインの計算に使われるライン位相情報d<sub>sk</sub>と、出力スキップラインの計算に使われるライン位相情報d<sub>sk</sub>の初期化を行う。ただし、リニアアレイ型多並列プロセッサ内での処理の遅延時間があるため、その遅延を補償するために上記ライン位相情報d<sub>sk</sub>は0で、ライン位相情報d<sub>sk</sub>は遅延時間に応じたオフセット位相を持たせることを行う。

【0301】ステップST202では、次のラインの1ライン分画素データのデータ入力が可能か否かの判別を行い、当該次ラインの画素データの入力が可能となるまで、この判断を繰り返す。当該ステップST202にて次ラインの画素データの入力が可能になると、次のステップST203で1ライン分の画素データの入力が行われる。

【0302】ステップST204では、後述する入力スキップライン制御信号F<sub>sk</sub>に従い、もしも入力スキップライン制御信号F<sub>sk</sub>が0ならば、ステップST205のように、ステップST203で入力した1ライン分のデータを補間元のデータが格納されているローカルメモリ上の4Hディレイライン56に格納する。逆に、入力スキップライン制御信号F<sub>sk</sub>が1ならば、このラインは不要とみなし、4Hディレイライン56には格納されず、そのデータは破壊され、ステップST206に進む。

【0303】この入力スキップライン制御信号F<sub>sk</sub>は、拡大走査線変換の際に意味をもつものであり、図32に拡大変換時の入力スキップライン制御信号F<sub>sk</sub>とフィールドメモリ51のデータとの関係を図示する。

【0304】図32の図中L<sub>sk</sub>がスキップされるラインを示しており、すなわち入力スキップライン制御信号F<sub>sk</sub>が1のときにはフィールドメモリ51の出力を止めると共にデータを取り込まないようにし、0の時にはメモリからのデータを取り込み、補間演算に必要な4Hディレイライン56の格納する。なお、入力スキップライン制御信号F<sub>sk</sub>の極性は1でスキップ、0でスキップしないとしているが、フィールドメモリの制御信号の局性が逆ならば反転する必要がある。

【0305】図28中の4Hディレイライン56の制御信号GRLは、いわゆるグローバルローテーションを表しており、1でそれぞれのラインデータを1ラインずつディレイし、0でディレイしないことを示す。これはライン単位の処理であるので、当該DSPにおいてはローカルメモリ内のデータを移動させることであるのでわずか数十ステップのインストラクションで済む。

【0306】以上の結果、信号用ディレイライン56には、キュービク補間に必要な4ラインのデータが得られることになる。例えば、図32の場合、図33に示すラインが上記信号用ディレイライン56のローカルメモ

りに格納される。

【0307】出力スキップライン制御信号  $F_{out}$  は、縮小走査線数変換の際に意味をもつものであり、図34に縮小変換時の出力スキップライン制御信号  $FOSL$  とフィールドメモリ64のデータとの関係を図示する。

【0308】図34の図中  $L_{out}$  がスキップされるラインを示しており、すなわち出力スキップライン制御信号  $F_{out}$  が1のときにはフィールドメモリ64への入力データを取り込まないようにし、0の時には出力信号  $OR$  のデータを取り込む。なお、出力スキップライン制御信号  $F_{out}$  の局性は1でスキップ、0でスキップしないとしているが、フィールドメモリの制御信号の局性が逆ならば反転する必要がある。この時のローカルメモリの内容を図35に示す。

【0309】ステップ  $ST206$  では出力スキップライ

$$Q_i = C_i((M+d_{out})/M)*d_{out} + C_i(d_{out}/M)*d_{out} + C_i((M-d_{out})/M)*d_{out} + C_i((2M-d_{out})/K)*d_{out}$$

ただし、 $C_1(x) = -|x|^3 + 5|x|^2 - 8|x| + 4$   
 $C_2(x) = |x|^3 - 2|x|^2 + 1$

出力スキップライン位相情報  $d_{out}$  は前ラインの出力スキップライン位相情報  $d_{out}$  が格納されたレジスタ60（出力スキップライン計算部60のレジスタ）を示しており、これについては後述する。 $d_{out}$ 、 $d_{out}$ 、 $d_{out}$ 、 $d_{out}$  は入力ラインをそれぞれ3H、2H、1H、0Hラインディレイさせたものである。

【0312】このときの各構成要素のローカルメモリの各レジスタの様子を図33に示す。出力スキップライン位相情報  $d_{out}$  は1ライン中でどの画素をとっても、ある一定の値になっている。

【0313】ステップ  $ST208$ 、ステップ  $ST209$ 、ステップ  $ST210$  では、次のラインの出力スキップラインであるかどうかの判定を行う。すなわち前のラインの出力スキップライン位相レジスタ  $d_{out}$  に格納された値に  $N$  を加算して、得られた値が  $M$  の2倍の値よりも大きい場合は1ビットのレジスタ  $F_{out}$  に1を格納し、出力ラインをスキップするものとし、逆の場合はレジスタ  $F_{out}$  に0を格納し、出力ラインをスキップさせないように設定する。

【0314】次の、ステップ  $ST211$ 、ステップ  $ST212$ 、ステップ  $ST213$  では、前記出力スキップラインを示すレジスタ  $F_{out}$  に従い、次のラインの出力スキップライン位相レジスタ  $d_{out}$  を計算する。もし、ステップ  $ST209$ 、ステップ  $ST210$  で計算したレジスタ  $F_{out}$  が1ならば、次ラインの出力スキップライン位相情報レジスタ  $d_{out}$  を現ラインの出力スキップライン位相情報レジスタ  $d_{out}$  から  $M$  を引いたものとし、そうでない場合には  $(N-M)$  を足したものにす。

【0315】ステップ  $ST214$ 、ステップ  $ST215$ 、ステップ  $ST216$  では、次のラインが入カスキ

ップライン位相情報  $d_{out}$  からキュービック係数を計算する。すなわち、式(1)を用いた計算を行う。ここで、ライン位相情報を表すものとしては、出力スキップライン位相情報  $d_{out}$  以外にもスキップライン位相情報  $d_{out}$  があるが、入力スキップライン位相情報  $d_{out}$  は実際の補間演算には使わず、以降に述べる入力スキップラインを決定するために使われる。このようにライン位相情報としては  $d_{out}$ 、 $d_{out}$  の2種類を持つが、実際の補間演算では出力スキップライン位相情報  $d_{out}$  のみを用いることで縮小から拡大までの任意比率変換を可能とする。

【0310】ステップ  $ST207$  では、上記ステップ  $ST206$  にて求めたキュービック係数と上記近傍4ラインの画素データの畳み込み演算により補間されるライン  $Q_i$  は以下の式(18)のようになる。

【0311】

$$\dots (18)$$

ラインであるかどうかの判定を行う。すなわち、前のラインの出力スキップライン位相情報レジスタ  $d_{out}$  に格納された値に  $N$  を加算して、得られた値が  $M$  の値と等しいか小さい場合は1ビットのレジスタ  $F_{out}$  に1を格納し、入力ラインをスキップするものとし、逆の場合はレジスタ  $F_{out}$  に0を格納し、入力ラインをスキップさせないように設定する。

【0316】次のステップ  $ST217$ 、ステップ  $ST218$ 、ステップ  $ST219$  では、前記入力スキップラインを示すレジスタ  $F_{out}$  に従い、次ラインの入カスキップライン位相レジスタ  $d_{out}$  を計算する。もし、ステップ  $ST214$ 、ステップ  $ST216$  で計算したレジスタ  $F_{out}$  が1ならば、次ラインの入カスキップライン位相情報レジスタ  $d_{out}$  を現ラインの入カスキップライン位相情報レジスタ  $d_{out}$  から  $M$  を引いたものとし、そうでなければそのままの値とする。

【0317】ステップ  $ST220$  にて、上記のように得られたキュービック補間演算後のデータと出力スキップライン制御信号  $F_{out}$  と入力スキップライン制御信号  $F_{out}$  を出力する。

【0318】ステップ  $ST220$  までが、1ライン分の計算であり、これを1フレームの間繰り返す。すなわち、ステップ  $ST222$  で1フレームの終わりかどうかを判断し、そうであれば図29のステップ  $ST201$  に戻り、そうでなければステップ  $ST202$  へ戻る。

【0319】以上説明したように、本発明実施の形態によれば、回路規模、その構成の複雑さからハードウェアでは実現困難なフィルタスイッチング補間方式（キュービックフィルタ補間方式）を用い、任意比率の画素数変換処理と走査線数変換処理をSIMD制御のリニアアレイ型多並列プロセッサDSPを使い、ソフトウェア処理だけで実現可能である。また、前記ASIC等のハード

ワイアードでは実現困難であった任意画素数への変換にも対応し、4:4:4フォーマット、4:2:2フォーマット、4:1:1フォーマットという任意のクロマフォーマットにも対応させることができる。その上、従来のハードワイアード回路と違い、各種フォーマットやビット精度についても全てソフトウェアの変更だけで対応できるため、外部に新たに回路を追加することがなくなる。

【0320】さらに、本発明の実施の形態によれば、リアルタイムに画素数及び走査線数をそれぞれ任意のサイズに変換することも可能である。

【0321】

【発明の効果】本発明においては、SIMD制御のリニアアレイ型多並列プロセッサを使い、ソフトウェア処理だけで、任意比率の画素数変換や走査線数変換のためのデジタル信号処理を実現可能とし、画素数、走査線数の変換比率はリアルタイムに設定可能で、それぞれ独立に設定できる。走査線数変換については、外部に置かれているフィールドメモリの制御信号をもリニアアレイ型多並列プロセッサにて計算することで、メモリコントロール回路を不要としている。また、本発明においては、例えば4:4:4フォーマットのみならず、4:2:2フォーマットや4:1:1フォーマットのような任意のクロマフォーマットに対応している。

【図面の簡単な説明】

【図1】リニアアレイ型多並列プロセッサの基本構成を示すブロック回路図である。

【図2】本発明実施の形態のリニアアレイ型多並列プロセッサの概略構成を示すブロック回路図である。

【図3】輝度信号についての拡大画素数変換処理の全体の流れを示すフローチャートである。

【図4】輝度信号についての拡大画素数変換処理における入力スキップレジスタの画素スキップ情報と位相情報の計算の流れを示すフローチャートである。

【図5】輝度信号についての拡大画素数変換処理における入力画素の近傍4画素の計算の前半部の流れを示すフローチャートである。

【図6】輝度信号についての拡大画素数変換処理における入力画素の近傍4画素の計算の後半部の流れを示すフローチャートである。

【図7】輝度信号の拡大画素数変換の動作説明に用いる図である。

【図8】輝度信号についての縮小画素数変換処理の全体の流れを示すフローチャートである。

【図9】輝度信号についての縮小画素数変換処理における出力スキップレジスタの画素スキップ情報と位相情報の計算の流れを示すフローチャートである。

【図10】輝度信号についての縮小画素数変換処理における入力画素の近傍4画素の計算の前半部の流れを示すフローチャートである。

【図11】輝度信号の縮小画素数変換の動作説明に用いる図である。

【図12】4:2:2フォーマットのクロマ信号の画素数変換処理の全体の流れを示すフローチャートである。

【図13】4:2:2フォーマットのクロマ信号の拡大画素数変換処理におけるフラグレジスタの計算の流れを示すフローチャートである。

【図14】4:2:2フォーマットのクロマ信号の拡大(縮小)画素数変換処理におけるクロマ信号の直線補間処理の前段の流れを示すフローチャートである。

【図15】4:2:2フォーマットのクロマ信号の拡大(縮小)画素数変換処理におけるクロマ信号の直線補間処理の中段の流れを示すフローチャートである。

【図16】4:2:2フォーマットのクロマ信号の拡大(縮小)画素数変換処理におけるクロマ信号の直線補間処理の後段の流れを示すフローチャートである。

【図17】4:2:2フォーマットのクロマ信号の拡大画素数変換処理におけるフラグレジスタの計算の流れを示すフローチャートである。

【図18】4:2:2フォーマットから4:4:4フォーマットに変換してクロマ信号の画素数変換処理を行う際の全体の流れを示すフローチャートである。

【図19】4:2:2フォーマットから4:4:4フォーマットへの変換の具体的な説明に用いる図である。

【図20】4:2:2フォーマットから4:4:4フォーマットへのフォーマット変換の流れを示すフローチャートである。

【図21】4:1:1フォーマットのクロマ信号の画素数変換処理を行う際の全体の流れを示すフローチャートである。

【図22】4:1:1フォーマットから4:2:2フォーマットへのフォーマット変換の流れの前段を示すフローチャートである。

【図23】4:1:1フォーマットから4:2:2フォーマットへのフォーマット変換の流れの中段を示すフローチャートである。

【図24】4:1:1フォーマットから4:2:2フォーマットへのフォーマット変換の流れの後段を示すフローチャートである。

【図25】4:1:1フォーマットの画素数変換時のローカルメモリの内容説明に用いる前半部の図である。

【図26】4:1:1フォーマットの画素数変換時のローカルメモリの内容説明に用いる後半部の図である。

【図27】4:1:1フォーマットから4:2:2フォーマットへのフォーマット変換時に使用するFIRフィルタの構成を示す回路図である。

【図28】任意比率の画素数変換処理と走査線数変換処理を行うための構成とリニアアレイ型多並列プロセッサ内部のブロック分けされた処理を示す図である。

【図29】走査線数変換処理の前段の流れを示すフロー

チャートである。

【図30】走査線数変換処理の中段の流れを示すフローチャートである。

【図31】走査線数変換処理の後段の流れを示すフローチャートである。

【図32】輝度信号の拡大ライン数変換の動作説明に用いる図である。

【図33】輝度信号の拡大ライン数変換の動作説明におけるローカルメモリの内容を示した図である。

【図34】輝度信号の縮小ライン数変換の動作説明に用

いる図である。

【図35】輝度信号の縮小ライン数変換の動作説明におけるローカルメモリの内容を示した図である。

【図36】2:3拡大画素数変換の原理説明に用いる図である。

【図37】キュービック関数の説明に用いる図である。

【図38】3:2縮小画素数変換の原理説明に用いる図である。

【図39】従来の画素数変換装置のハードウェア構成を示すブロック回路図である。

【図40】従来のハードウェア構成による画素数変換装置における2:3拡大画素数変換の動作説明に用いる図である。

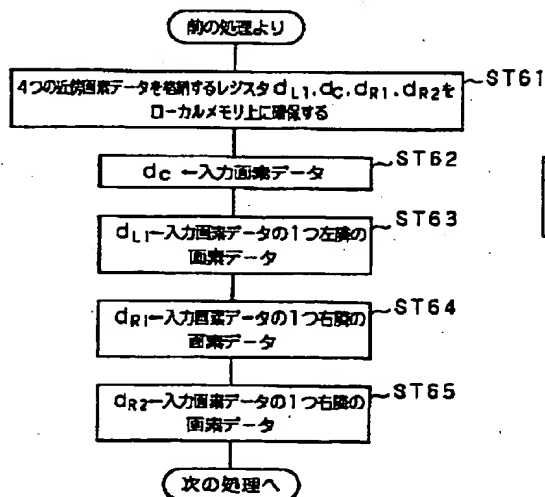
【図41】従来のハードウェア構成による画素数変換装置における3:2縮小画素数変換の動作説明に用いる図である。

【図42】2:3拡大ライン数変換の原理説明に用いる図である。

【図43】3:2縮小ライン数変換の原理説明に用いる図である。

【図44】従来の走査線数変換装置のハードウェア構成を示すブロック回路図である。

【図10】



【図45】従来のハードウェア構成による走査線数変換装置における2:3拡大ライン数変換の動作説明に用いる図である。

【図46】従来のハードウェア構成による走査線数変換装置における3:2縮小ライン数変換の動作説明に用いる図である。

【図47】4:4:4フォーマット構造を示す図である。

【図48】4:2:2フォーマット構造を示す図である。

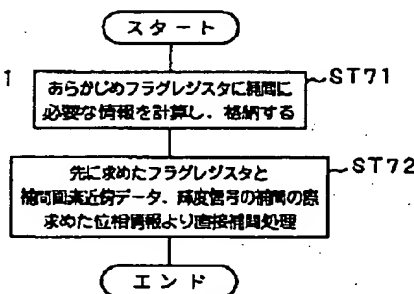
【図49】4:1:1フォーマット構造を示す図である。

【図50】4:1:1フォーマットから4:2:2フォーマットへの変換を示す図である。

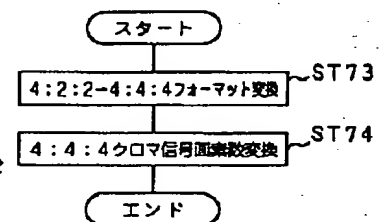
【符号の説明】

- 1 リニアアレイ型多並列プロセッサ、 10 要素プロセッサ、 11 入力レジスタ、 12 入力スキップレジスタ、 13 出力レジスタ、 14 出力スキップレジスタ、 15 ローカルメモリ、 16 演算処理部、 50 データ入力端子、 51 フィールドメモリ、 52 縮小画素数変換用出力スキップ画素計算部、 53 拡大画素数変換用入力スキップ画素計算部、 54 画素数キュービック補間演算処理部、 55 画素数変換用画素位相計算部、 56 ディレイライン、 57 走査線数変換用キュービック補間演算処理部、 58、 縮小走査線数変換用ライン位相計算部、 59 ライン位相用レジスタ、 60 出力スキップライン計算部、 61 出力スキップライン用レジスタ、 62 拡大走査線数変換用ライン位相計算部、 63 入力スキップライン計算部、 64 フィールドメモリ、 65 データ出力端子、 71 画素数変換処理部、 72 リニアアレイ型多並列プロセッサ

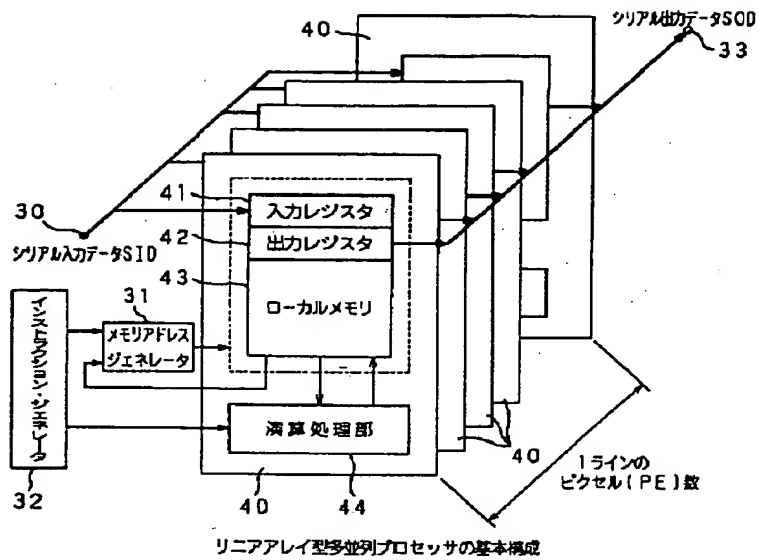
【図12】



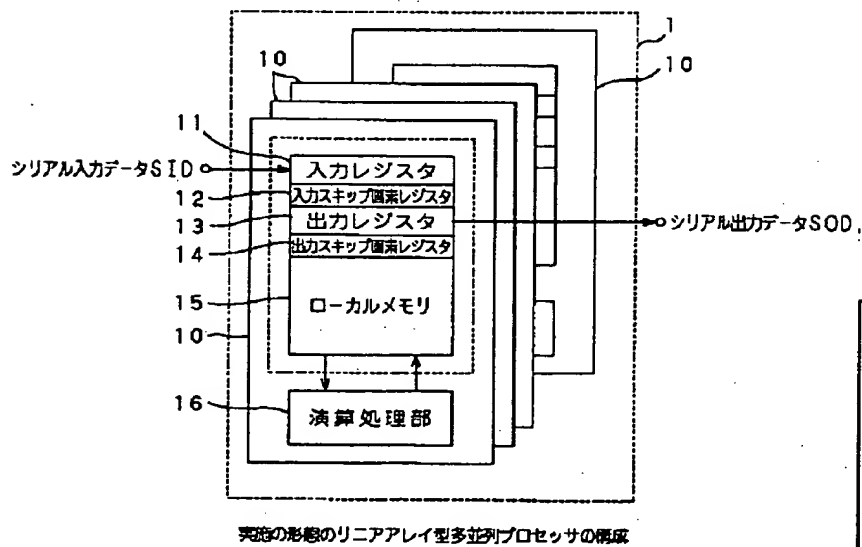
【図18】



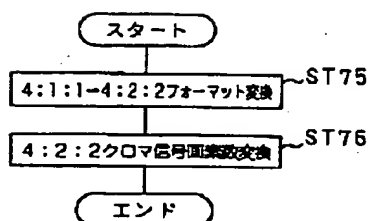
【図 1】



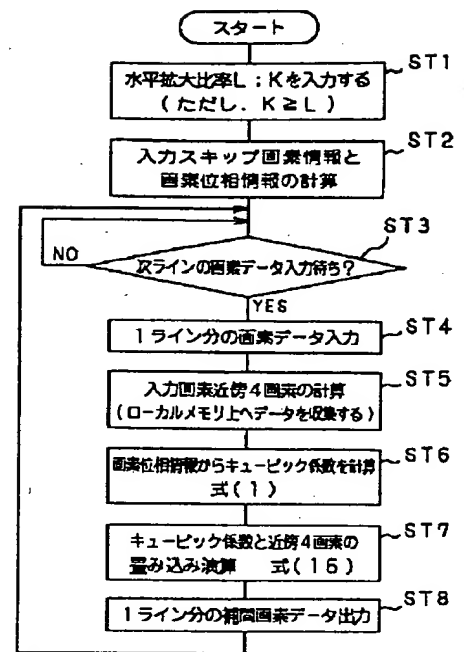
【図 2】



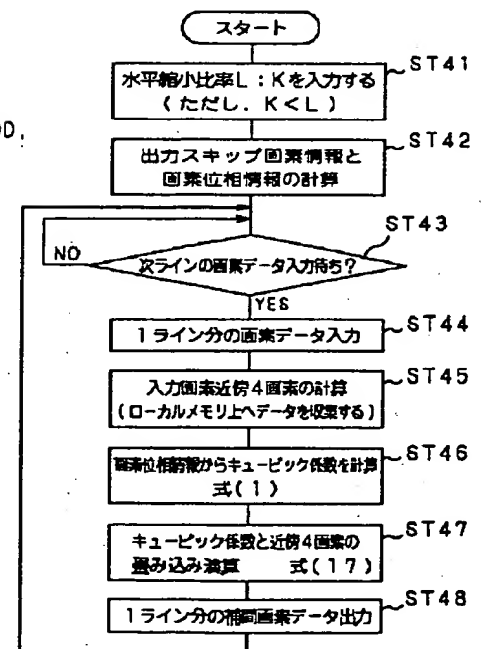
【図 21】



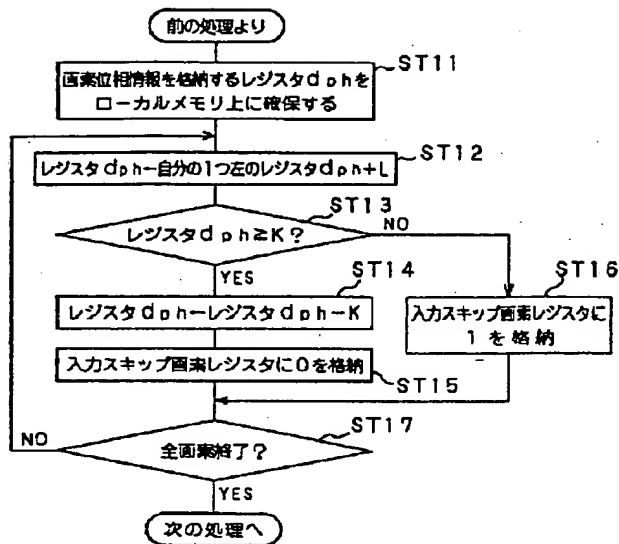
【図 3】



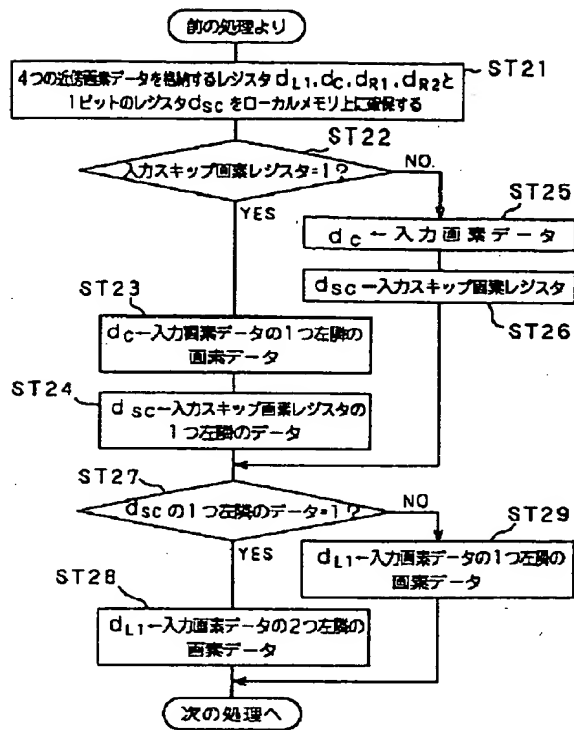
【図 8】



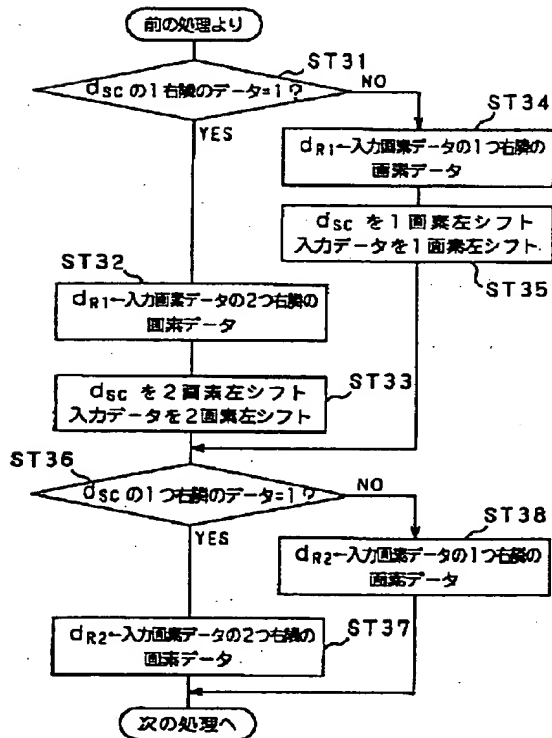
【図 4】



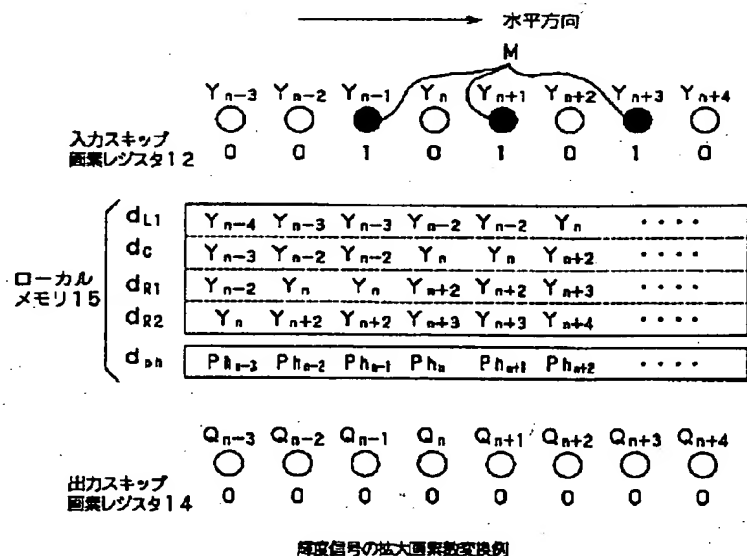
【図 5】



【図 6】



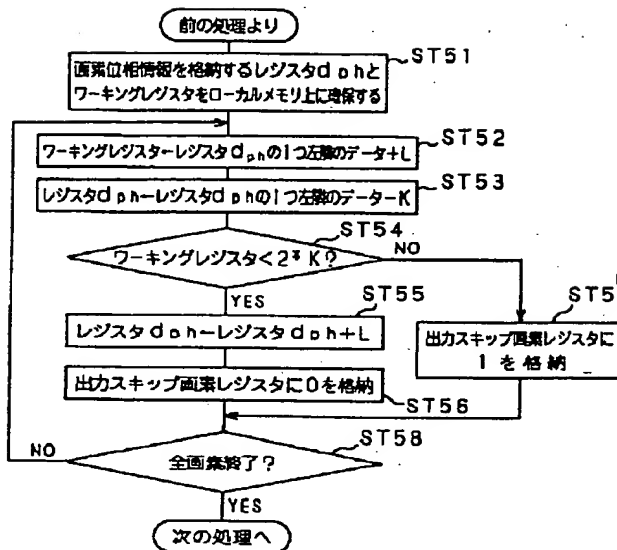
【図 7】



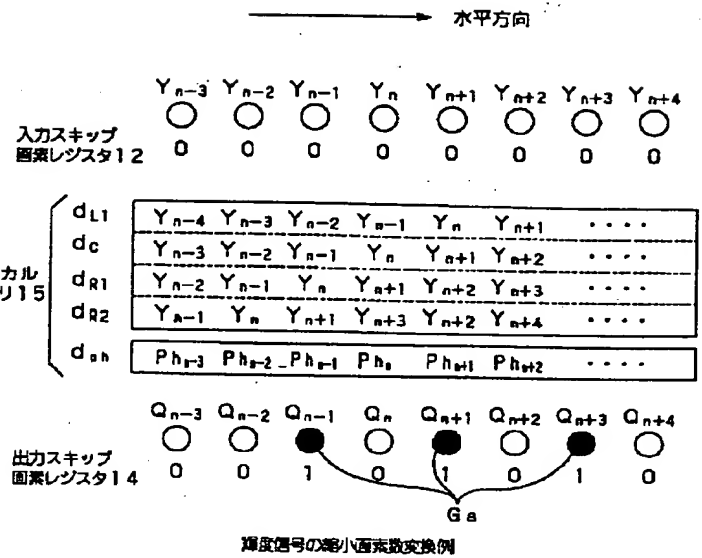
図素データの拡大図素データ変換例



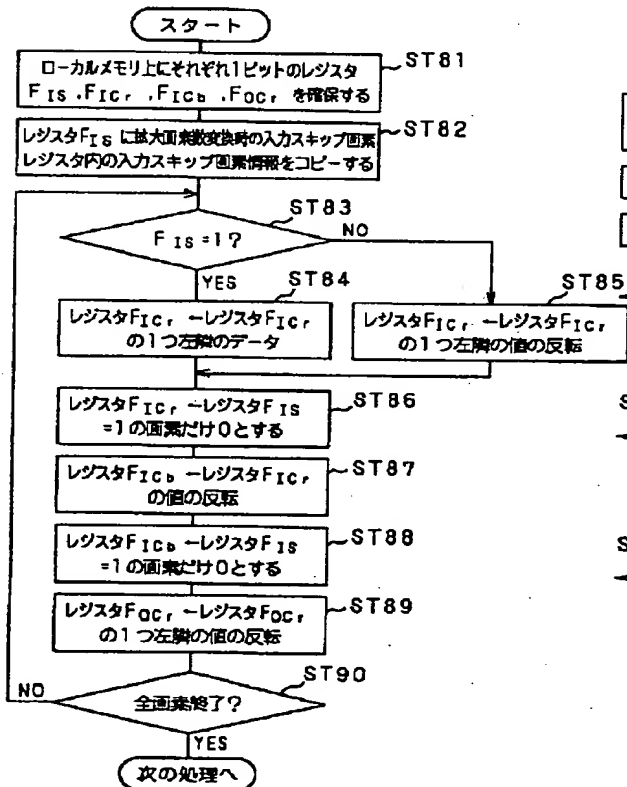
【図 9】



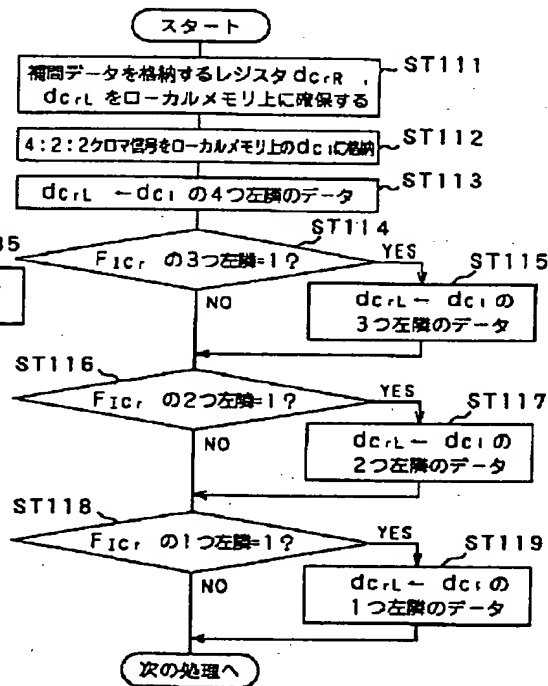
【図 11】



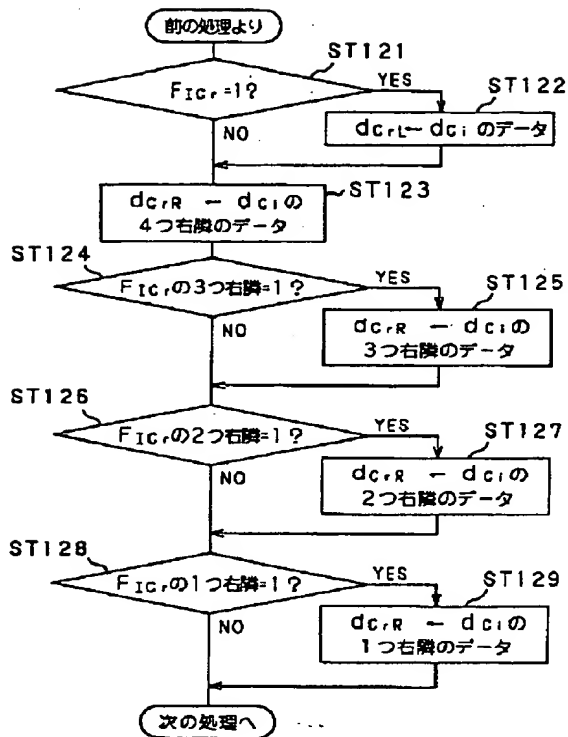
【図 13】



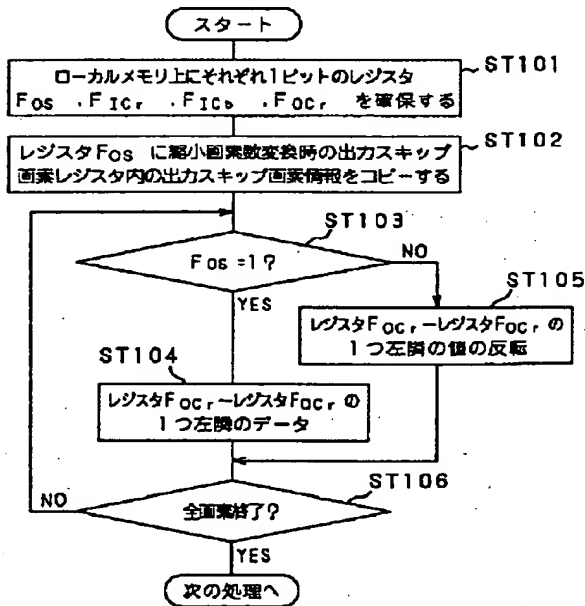
【図 14】



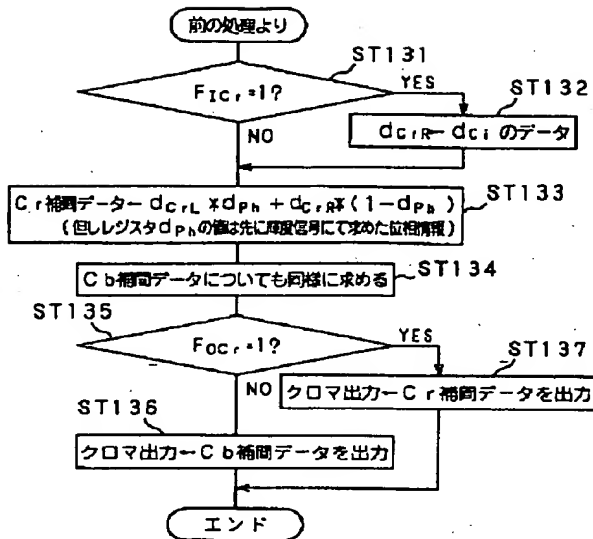
【図15】



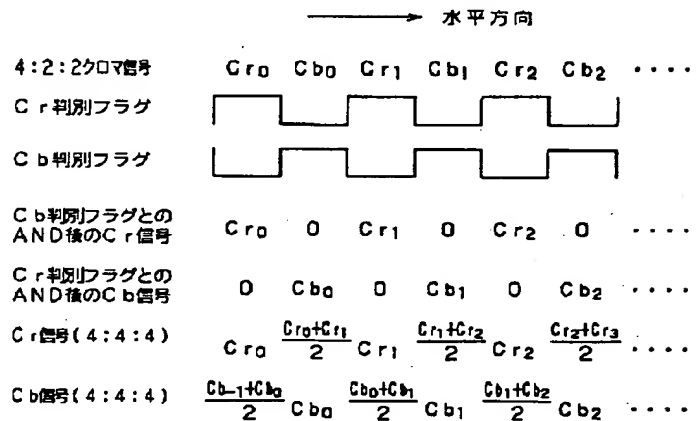
【図17】



【図16】

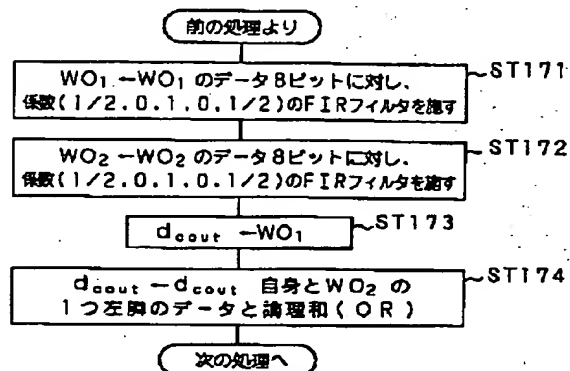


【図19】

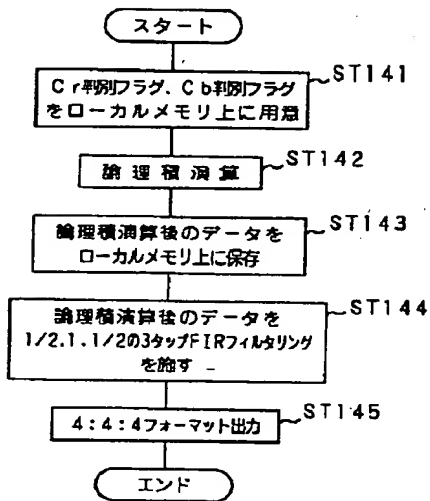


4:2:2:-4:1:1クロマフォーマット変換例

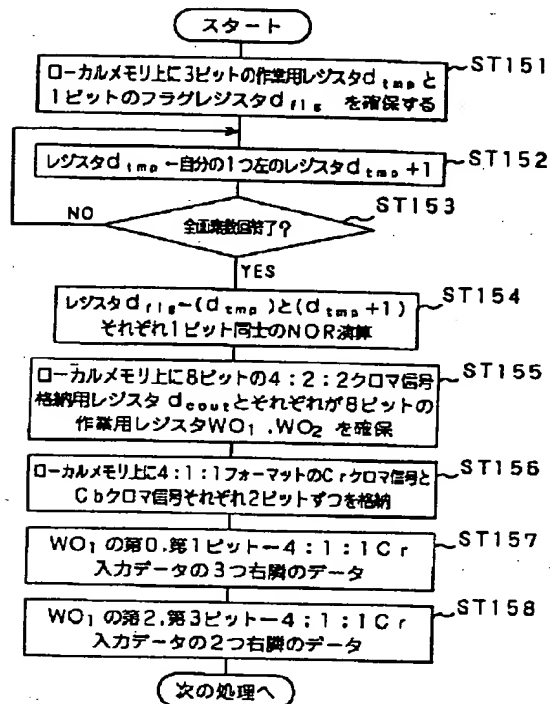
【図24】



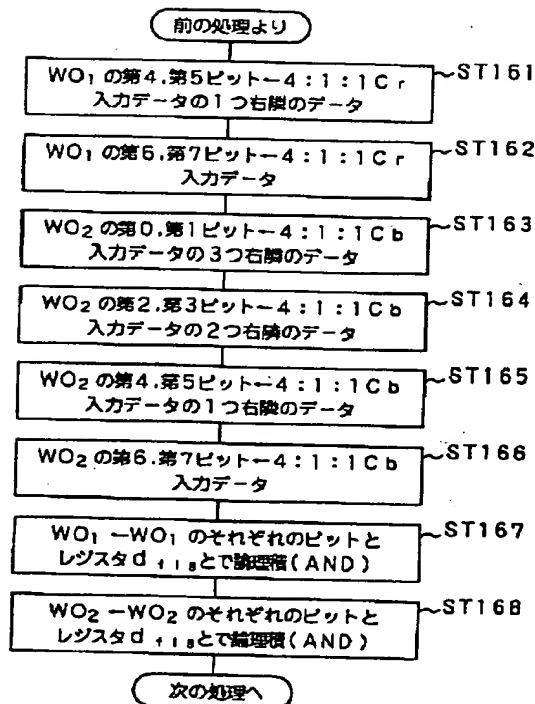
【図 20】



【図 22】



【図 23】



【図 25】

Cr(2bit)		Cr07	Cr05	Cr03	Cr01	Cr17
		Cr06	Cr04	Cr02	Cr00	Cr16
判別フラグ信号						
(a) レジスタ WO1	WO1+7	Cr07	Cb05	Cr03	Cb01	Cr17
	WO1+6	Cr06	Cb04	Cr02	Cb00	Cr16
	WO1+5	Cr05	Cb03	Cr01	Cb17	Cr15
	WO1+4	Cr04	Cb02	Cr00	Cb16	Cr14
	WO1+3	Cr03	Cb01	Cr07	Cb05	Cr13
	WO1+2	Cr02	Cb00	Cr06	Cb04	Cr12
	WO1+1	Cr01	Cb17	Cr15	Cb13	Cr11
	WO1	Cr00	Cb16	Cr14	Cb12	Cr10
(b) レジスタ WO1	WO1+7	Cr07	0	0	0	Cr17
	WO1+6	Cr06	0	0	0	Cr16
	WO1+5	Cr05	0	0	0	Cr15
	WO1+4	Cr04	0	0	0	Cr14
	WO1+3	Cr03	0	0	0	Cr13
	WO1+2	Cr02	0	0	0	Cr12
	WO1+1	Cr01	0	0	0	Cr11
	WO1	Cr00	0	0	0	Cr10
(c) レジスタ WO1	WO1+7	Cr07	0	$\frac{Cb07+Cb17}{2}$	0	Cr17
	WO1+6	Cr06	0	$\frac{Cb06+Cb16}{2}$	0	Cr16
	WO1+5	Cr05	0	$\frac{Cb05+Cb15}{2}$	0	Cr15
	WO1+4	Cr04	0	$\frac{Cb04+Cb14}{2}$	0	Cr14
	WO1+3	Cr03	0	$\frac{Cb03+Cb13}{2}$	0	Cr13
	WO1+2	Cr02	0	$\frac{Cb02+Cb12}{2}$	0	Cr12
	WO1+1	Cr01	0	$\frac{Cb01+Cb11}{2}$	0	Cr11
	WO1	Cr00	0	$\frac{Cb00+Cb10}{2}$	0	Cr10

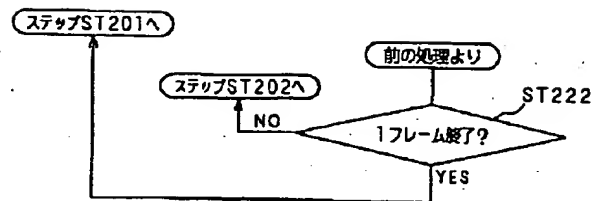
【図 26】

判別フラグ番号

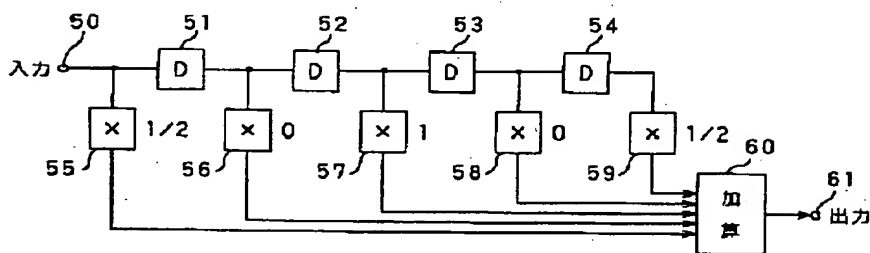
$d_{out} + 7$	$Cr_{07}$	$Cb_{07}$	$\frac{Cb_{07} + Cb_{17}}{2}$	$\frac{Cr_{07} + Cr_{17}}{2}$	$Cr_{17}$
$d_{out} + 6$	$Cr_{06}$	$Cb_{06}$	$\frac{Cb_{06} + Cb_{16}}{2}$	$\frac{Cr_{06} + Cr_{16}}{2}$	$Cr_{16}$
$d_{out} + 5$	$Cr_{05}$	$Cb_{05}$	$\frac{Cb_{05} + Cb_{15}}{2}$	$\frac{Cr_{05} + Cr_{15}}{2}$	$Cr_{15}$
$d_{out} + 4$	$Cr_{04}$	$Cb_{04}$	$\frac{Cb_{04} + Cb_{14}}{2}$	$\frac{Cr_{04} + Cr_{14}}{2}$	$Cr_{14}$
$d_{out} + 3$	$Cr_{03}$	$Cb_{03}$	$\frac{Cb_{03} + Cb_{13}}{2}$	$\frac{Cr_{03} + Cr_{13}}{2}$	$Cr_{13}$
$d_{out} + 2$	$Cr_{02}$	$Cb_{02}$	$\frac{Cb_{02} + Cb_{12}}{2}$	$\frac{Cr_{02} + Cr_{12}}{2}$	$Cr_{12}$
$d_{out} + 1$	$Cr_{01}$	$Cb_{01}$	$\frac{Cb_{01} + Cb_{11}}{2}$	$\frac{Cr_{01} + Cr_{11}}{2}$	$Cr_{11}$
$d_{out}$	$Cr_{00}$	$Cb_{00}$	$\frac{Cb_{00} + Cb_{10}}{2}$	$\frac{Cr_{00} + Cr_{10}}{2}$	$Cr_{10}$

レジスタ  $d_{out}$

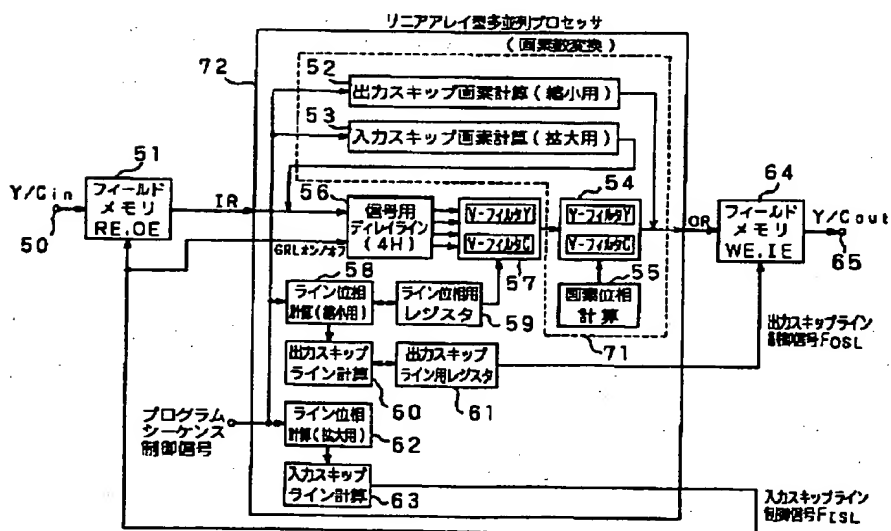
【図 31】



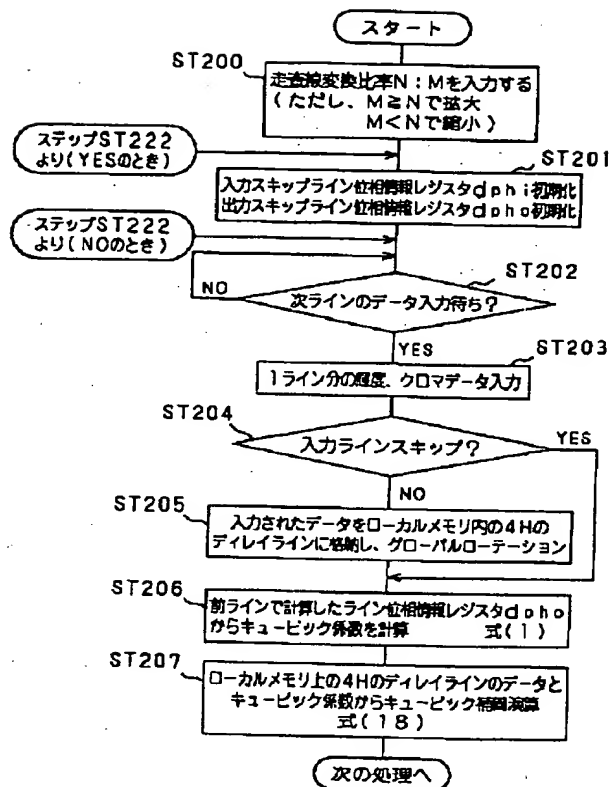
【図 27】



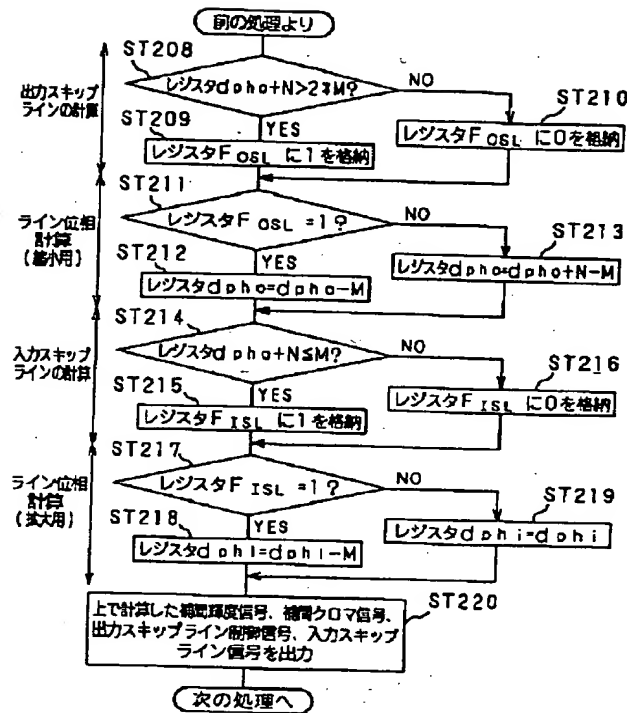
【図 28】



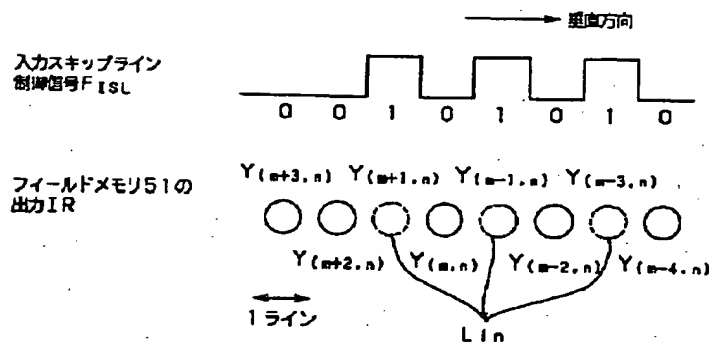
【図29】



【図30】

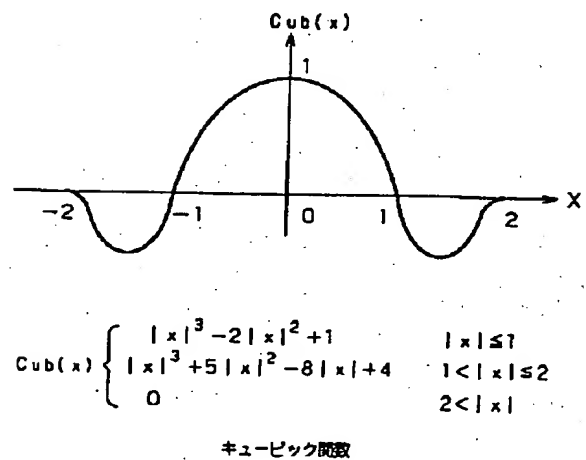


【図32】

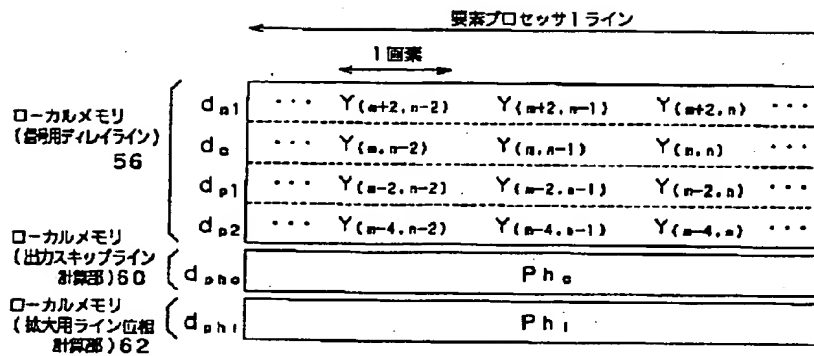


輝度信号の拡大走査数変換例(その1)

【図37】



【図 33】

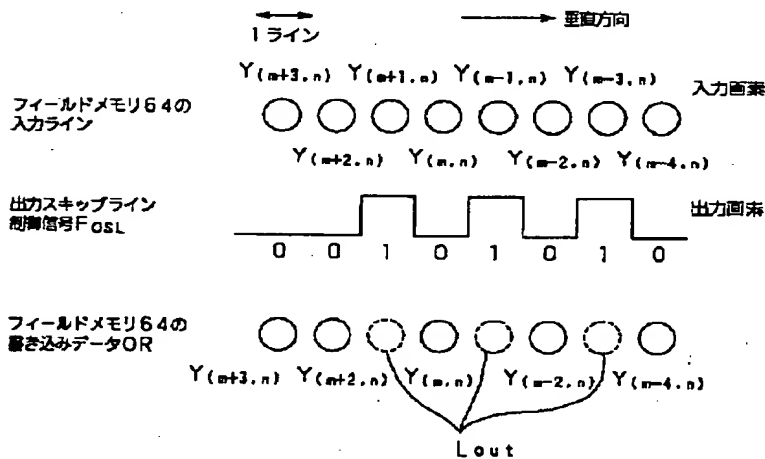


輝度信号の拡大走査線変換例 (その2)

【図 48】

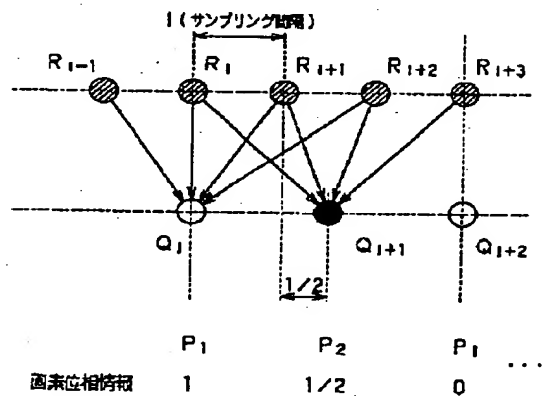
タイムインデックス	n	n+1	n+2	n+3
Y(8bit)	Y07	Y17	Y27	Y37
	Y08	Y18	Y28	Y38
	Y05	Y16	Y26	Y36
	Y04	Y15	Y25	Y35
	Y03	Y14	Y24	Y34
	Y02	Y13	Y23	Y33
	Y01	Y12	Y22	Y32
	Y00	Y11	Y21	Y31
	Y00	Y10	Y20	Y30
	Y00	Y10	Y20	Y30
Cb/Cr(8bit)	Cr07	Cb07	Cr27	Cb27
	Cr08	Cb08	Cr28	Cb28
	Cr05	Cb05	Cr26	Cb26
	Cr04	Cb04	Cr25	Cb25
	Cr03	Cb03	Cr24	Cb24
	Cr02	Cb02	Cr23	Cb23
	Cr01	Cb01	Cr22	Cb22
	Cr00	Cb00	Cr21	Cb21
	Cr00	Cb00	Cr20	Cb20
	Cr00	Cb00	Cr20	Cb20

【図 34】



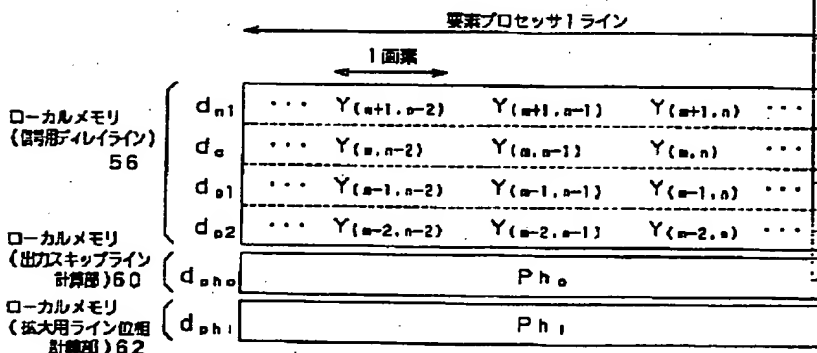
輝度信号の縮小走査線変換例 (その1)

【図 38】



3:2画素数変換 (縮小変換) 原理図

【図 35】

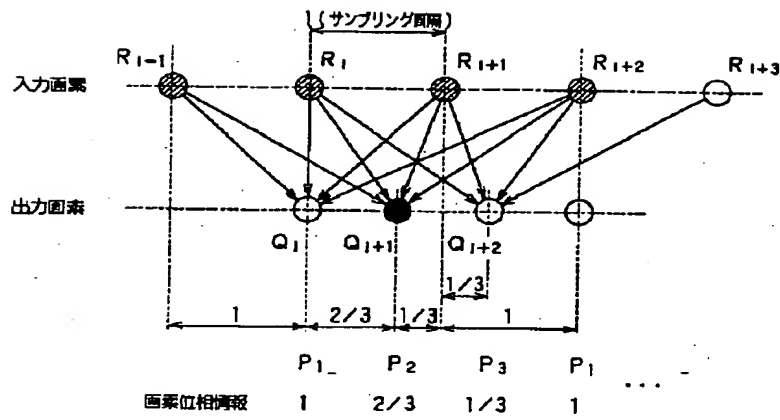


輝度信号の縮小走査線変換例 (その2)

【図 49】

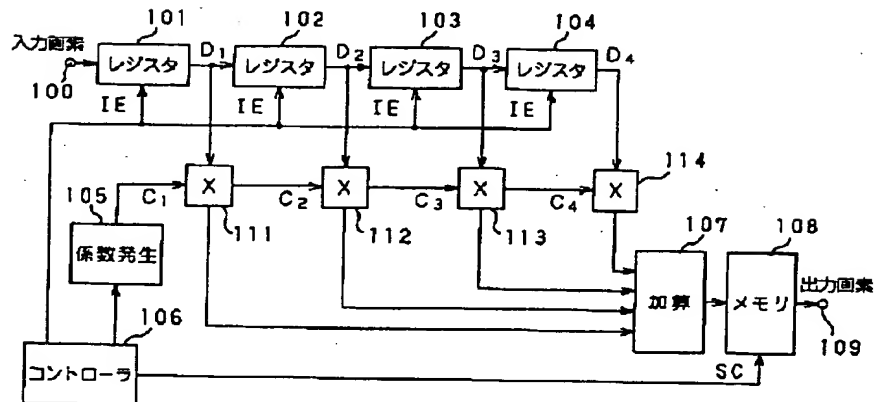
タイムインデックス	n	n+1	n+2	n+3
Y(8bit)	Y07	Y17	Y27	Y37
	Y06	Y16	Y26	Y36
	Y05	Y15	Y25	Y35
	Y04	Y14	Y24	Y34
	Y03	Y13	Y23	Y33
	Y02	Y12	Y22	Y32
	Y01	Y11	Y21	Y31
	Y00	Y10	Y20	Y30
	Y00	Y10	Y20	Y30
	Y00	Y10	Y20	Y30
Cr(2bit)	Cr07	Cr05	Cr03	Cr01
	Cr08	Cr04	Cr02	Cr00
	Cb07	Cb05	Cb03	Cb01
	Cb08	Cb04	Cb02	Cb00
	Cb07	Cb05	Cb03	Cb01
	Cb08	Cb04	Cb02	Cb00
	Cb07	Cb05	Cb03	Cb01
	Cb08	Cb04	Cb02	Cb00
	Cb07	Cb05	Cb03	Cb01
	Cb08	Cb04	Cb02	Cb00

【図 36】



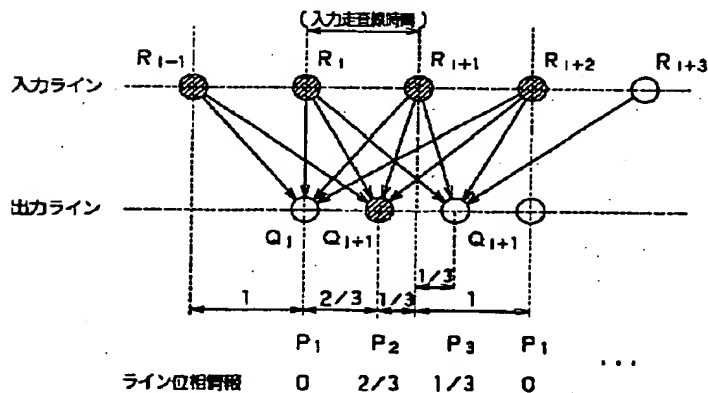
2:3要素数変換(拡大変換)

【図 39】



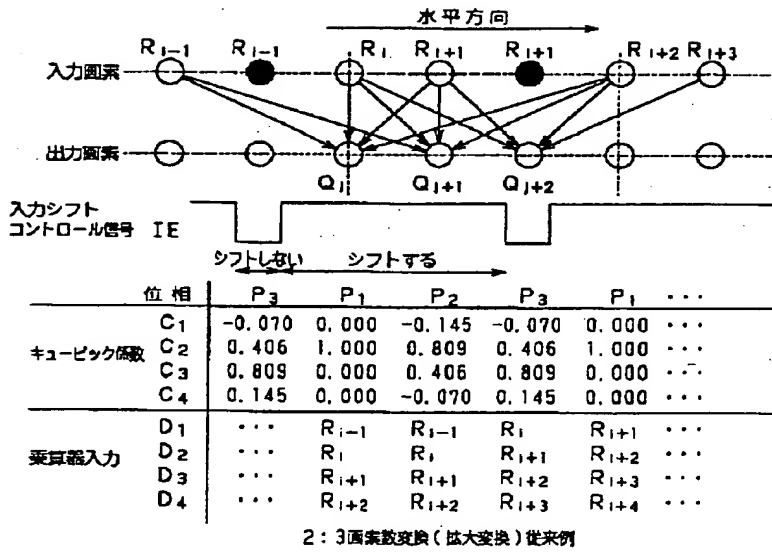
要素数変換処理のハードウェア構成図

【図 42】



3:2ライン数変換(拡大変換)

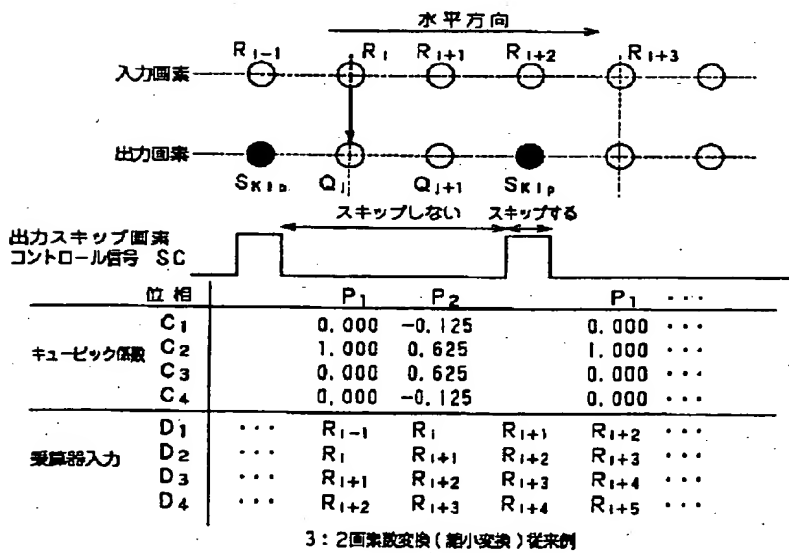
【図 40】



【図 47】

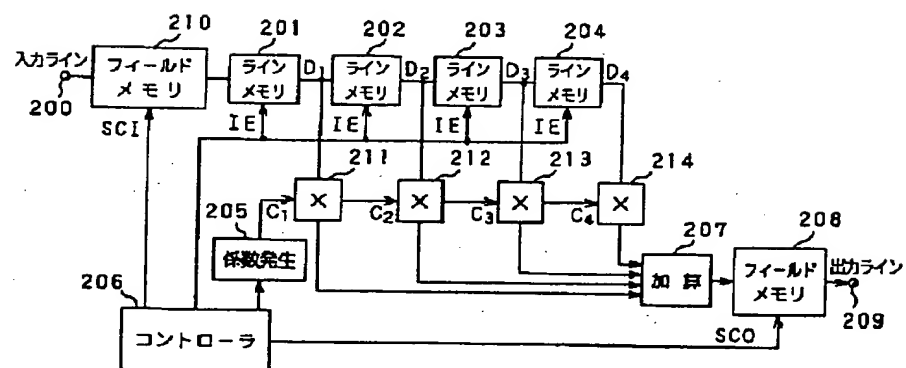
タイムインデックス	n	n+1	n+2	n+3
Y(8bit)	Y07	Y17	Y27	Y37
	Y08	Y18	Y28	Y38
	Y05	Y15	Y25	Y35
	Y04	Y14	Y24	Y34
	Y03	Y13	Y23	Y33
	Y02	Y12	Y22	Y32
	Y01	Y11	Y21	Y31
	Y00	Y10	Y20	Y30
	C r07	C r17	C r27	C r37
	C r08	C r18	C r28	C r38
Cr(8bit)	C r06	C r16	C r26	C r36
	C r04	C r14	C r24	C r34
	C r03	C r13	C r23	C r33
	C r02	C r12	C r22	C r32
	C r01	C r11	C r21	C r31
	C r00	C r10	C r20	C r30
	C b07	C b17	C b27	C b37
	C b06	C b16	C b26	C b36
	C b05	C b15	C b25	C b35
	C b04	C b14	C b24	C b34
Cb(8bit)	C b03	C b13	C b23	C b33
	C b02	C b12	C b22	C b32
	C b01	C b11	C b21	C b31
	C b00	C b10	C b20	C b30

【図 41】



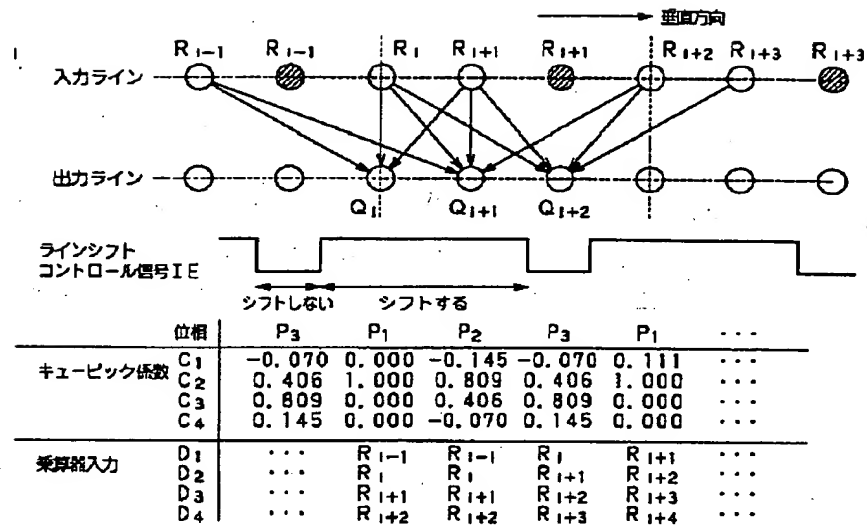


【图 4 4】



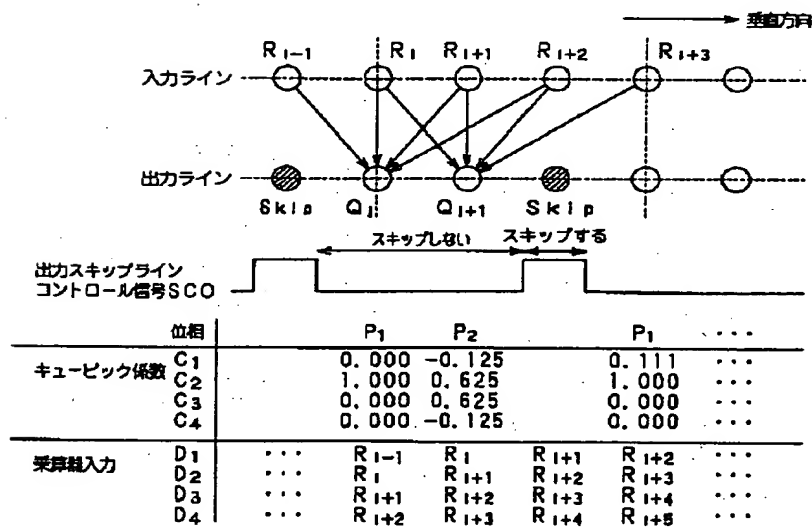
走査線数変換処理のハードウェア構成図

【図 45】



3:2ライン数変換(拡大変換)従来例

【図 46】



3:2ライン数変換(縮小変換)従来例

【図 50】

タイムインデックス		n	n+1	n+2	n+3
4:1:1	Cr(2bit)	Cr07	Cr05	Cr03	Cr01
		Cr06	Cr04	Cr02	Cr00
	Cb(2bit)	Cb07	Cb05	Cb03	Cb01
		Cb06	Cb04	Cb02	Cb00
4:2:2	C422(8bit)	Cr07	Cb07	Cr07	Cb07
		Cr06	Cb06	Cr06	Cb06
		Cr05	Cb05	Cr05	Cb05
		Cr04	Cb04	Cr04	Cb04
		Cr03	Cb03	Cr03	Cb03
		Cr02	Cb02	Cr02	Cb02
		Cr01	Cb01	Cr01	Cb01
		Cr00	Cb00	Cr00	Cb00

---

フロントページの続き

(51) Int. Cl.

9/64

識別記号

庁内整理番号

F I

G06F 15/66

技術表示箇所

355

D

(72) 発明者 中村 憲一郎

東京都品川区北品川 6 丁目 7 番 3 5 号 ソ

ニー株式会社内